An Arduino-Based Subsystem for Controlling UAVs Through GSM

George N. Solidakis, Fanourios M. Tsokas, Michael C. Batistatos, Nikos C. Sagias, George V. Tsoulos, Dimitra A. Zarbouti, Georgia E. Athanasiadou
Department of Informatics and Telecommunications, University of Peloponnese, Tripolis, Greece
(e-mail: {yorgossolidakis, fanouris.tsokas}@gmail.com, {mbatist, nsagias, gtsoulos, dzarb, gathanas}@uop.gr)

***Abstract*—Long distance and beyond sight of view communication between a ground control station and an Unmanned Aerial Vehicle (UAV) is mainly achieved using high-gain antennas, antenna trackers or satellites. These implementations demand expensive, heavy and not easily deployed equipment, while in many cases cannot provide a sufficient communication range. The current work proposes and demonstrates a lightweight and low-cost Arduino-based telecommunication subsystem that is capable of sending control commands to the UAV based on GSM or GPRS the most widely deployed cellular networks standard. During field trials control commands successfully transferred from a mobile phone as well as a laptop to the UAV autopilot, with an average time of 2.6 sec and 0.5 sec respectively. The proposed subsystem can be embedded independently or coexist with other control systems, contributing to a ubiquitous UAVs management system.**

***Index Terms*—Arduino, GSM network, long distance communications, UAV**

I. INTRODUCTION

The control communication link between the user and the Unmanned Aerial Vehicle (UAV) is one of the most important aspects of a UAV control system. The short range control commands transmission is mostly based on small transceivers, mainly operating at 2.4 GHz, with an effective range no more than 2km in Line-of-Sight (LoS) conditions. For longer communication distances and beyond pilot's sight, high-gain antennas are used, combined with antenna tracker systems [1] or even satellites [2]. These implementations are expensive and utilize heavy and not easily deployed equipment, while the communication range is limited. Trying to improve the characteristics and the performance of the Ground Control Station (GCS)-to-UAV communications, researchers and companies across the world have started to focus on exploiting the cellular network infrastructure. Delair-Tech, a French UAV manufacturer, recently announced a cooperation with Telefonica Business Solutions, a communication network provider, for connecting UAVs to the 3G network and providing beyond visual LoS communications [3]. Skydrone, a First Person View (FPV) system provider, has developed a system for obtaining video stream and telemetry data from the UAV through the 3G/4G network, however without supporting control commands [4]. The number of publications in the open literature regarding proposed systems for transmitting and receiving data from the UAV using features of the currently available cellular networks is very limited. In [5] a system is implemented and evaluated for controlling the UAV and obtaining telemetry data remotely, using the 3G network. In [6] a remote control transmitter is connected to a laptop and the GSM network is used for real time UAV control. Authors in [7] propose and simulate a UAV teleoperation virtual environment for controlling the UAV through GSM and getting flight conditions feedback, for enhancing the situational awareness of the remote pilot. In [8] a UAV based surveillance system is presented, where the video is streamed over the LTE network.

All the aforementioned contributions suggest telecommunication subsystems for either transmitting only telemetry data or controlling the UAV through the cellular network, but only in real time (e.g. turn left, go backwards etc.) using a remote control transmitter. This paper proposes and demonstrates the implementation of a lightweight and low-cost UAV control subsystem, which is mission oriented (e.g. waypoints insertion, return to base command etc.), aiming to a ubiquitous UAV (or even a fleet of distributed UAVs) mission management at long distances and beyond the visual line of sight, through the GSM/GPRS cellular network.

The most significant requirement for such an operation is the sufficient network coverage rather than the available bandwidth. Hence the GSM/GPRS network is an attractive solution due to its global coverage. The subsystem is based on Arduino platforms and components for converting the GSM/GPRS data to the appropriate UAV communication protocol and forwarding it to the UAV autopilot. This paper is structured as follows: Section II describes the subsystem architecture, while section III provides the corresponding implementation strategy, section IV discusses subsystem performance through field trials and finally section V concludes the paper.

II. SUBSYSTEM DESCRITPION

In this section an overview of the proposed subsystem is presented. The network architecture is introduced, the equipment is described and the control commands interface is explained.

A. Network Architecture

The generic network architecture is presented in Fig. 1. The GCS can be either a mobile phone that sends commands to the UAV as a simple text message (aka SMS) through the GSM network or any other device connected to the internet (e.g. laptop, tablet, etc.). In the latter case, the UAV receives the control commands through the internet using GPRS.

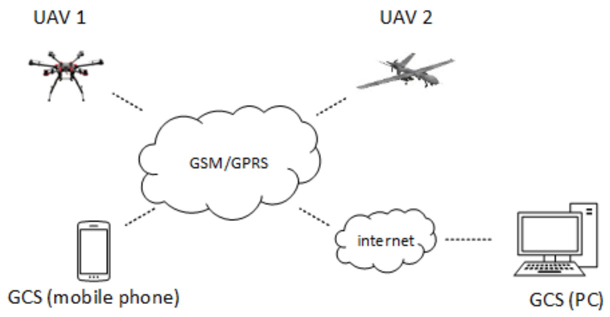

Fig. 1. Network Architecture.

The GCS can also transmit control commands to multiple UAVs, wherever a GSM/GPRS cellular network coverage is available, even if the UAVs are in distant territories or different countries by exploiting roaming services.

B. Equipment Description

Fig. 2 shows the UAV and the electronic devices that were used:

The UAV is an IRIS+ quadcopter by 3DRobotics. The navigation system is based on the Pixhawk autopilot, an open source hardware that is capable of autonomously accomplishing missions based on preloaded data as well as recording and transmitting telemetry and flight data. The microcontroller board is an Arduino Mega ADK Rev. 3. The programming of the board is done using the Arduino Integrated Development Environment (IDE). The GSM/GPRS module is an Arduino GSM shield with Quectel M10 modem. Utilizing a GSM antenna and a SIM card it can make or receive voice calls, send or receive text messages or connect to the internet.

The voltage regulator is Bi-directional logic level converter, which steps down or up the voltage between 5V and 3.3V.

Mission Planner was used as a ground control software. It is an open source software that provides a graphical interface to the user for communicating with the UAV.

C. Control Commands Interface

The interface between the GCS and the autopilot is comprised of three stages. The Arduino GSM shield, the Arduino Mega microcontroller and the voltage converter circuit (Fig. 3). The GCS sends the commands using the GSM/GPRS network and the GSM shield receives the signal. Arduino Mega, the main microcontroller of the communication subsystem, extracts the data from the GSM shield, processes it and forwards the resulting data to the UAV autopilot. The autopilot then executes the received command and also sends telemetry data back to the GCS following the reverse procedure.

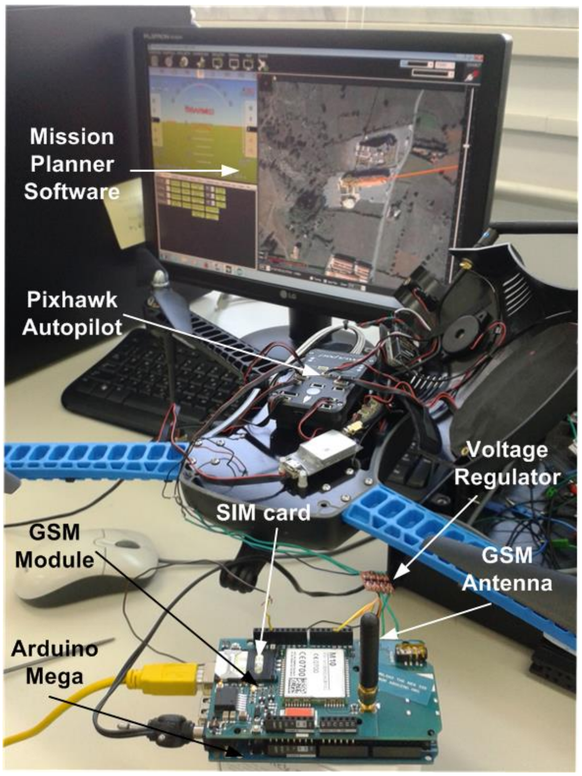

Fig. 2. Equipment.

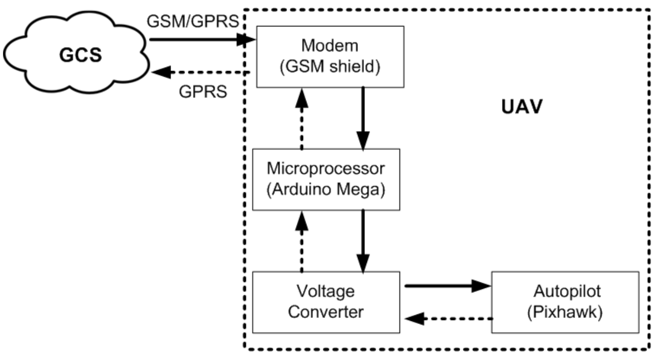

Fig. 3. Control Commands Process and Interface

III. IMPLEMENTATION AND COMMAND TYPES DETAILS

This section describes the subsystem implementation process and analyses the transmitted command types with examples.

A. The Arduino Platform

Fig. 4 shows the necessary connections between the GSM shield and the Arduino Mega board. Specifically, the GSM Tx (output D2) and Rx (input D3) of the GSM shield are connected to the Rx1 (input D19) and Tx1 (output D18) of the Arduino Mega board, respectively.

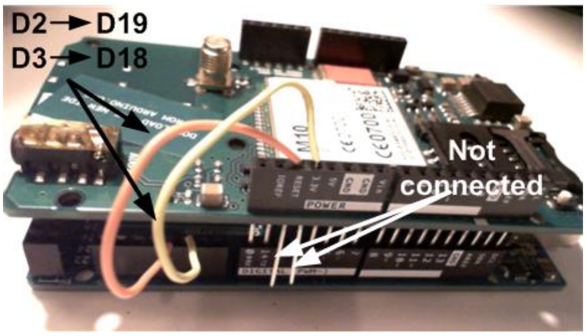

Fig. 4. Arduino platform.

Fig. 5 provides a schematic design of all the connections between the different system units. The Arduino Mega is connected to the Telemetry port 1 of the Pixhawk autopilot. A voltage reduction circuit is needed between the two devices, because the Arduino board outputs 5 Volt, while the autopilot port accepts 3.3 Volt. The Rx2 (input D17), Tx2 (output D16) and GND of the Arduino board are connected to the high voltage side of the voltage regulator and the autopilot telemetry port is connected to the low voltage side.

The Arduino Mega board runs also the necessary code for processing and managing the data from the GSM shield to the autopilot. The code that was developed for the purposes of this work can be found in [9].

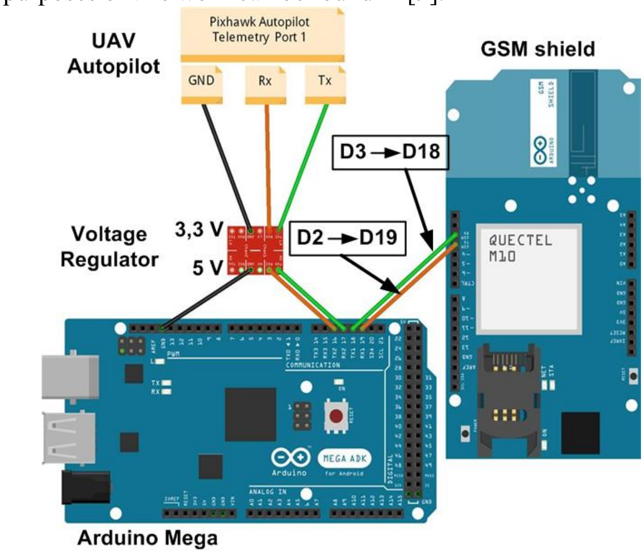

Fig. 5. System Schematic.

B. Command Types

The Pixhawk autopilot uses the Micro Air Vehicle Link (MAVLink) protocol [10] for communicating with the GCS, which is a widely used communication standard for a lot of commercial autopilots. When the user uses a GCS software that is connected to the internet, the system creates a UDP bridge between the autopilot and the GCS. The Arduino board is preconfigured with the GCS IP and port and MAVLink messages are being directed from the GCS to the UAV and vice versa, using GPRS. In this case, the exchanged data is originally organized in MAVLink packets, hence, no further processing is needed from the Arduino microcontroller. The latter one only connects the GSM/GPRS modem to the Pixahawk autopilot and manages the data traffic.

In the case of a user sending control commands through text messages, the operation includes more steps than before. Prior to sending the text messages that include the control commands, a message with a unique code (serving as a password) is sent to the Arduino platform for security reasons. If the code is valid the number of the mobile phone SIM card is being saved in the Arduino memory and the control commands that this mobile phone sends are being accepted by the system. This way, the UAV is safe from receiving control commands from an unknown user.

A simple text message that a user sends from his mobile phone does not follow the MAVLink protocol. Therefore, the Arduino Mega is preconfigured with a number of commands, organized as MAVLink packets that the UAV is capable to perform (e.g. return to base, land, take off, fly to a predetermined location etc.) and a unique number is assigned to each command. The user can choose one of these predefined commands or to send new waypoints. In the first case, the text message includes the number that has been assigned to the predefined command.

The user has also the option to send a sequence of commands that will be serially executed, by sending multiple text messages. In this case, the text messages also include redundancy data that sets the Arduino subsystem into stand-by mode and the commands sequence is being forwarded to the autopilot when all the messages have been received. Due to the redundancy data, the system is also capable of requesting missing messages or re-arranging them into the correct order (the system may receive the messages in a different order due to the network latency).

The user can also send new waypoints that do not exist in the Arduino memory. In this case, the text message includes the decimal degrees of the latitude and longitude and the elevation of the geographic coordinate system of the new waypoint (x, y, z).

Fig. 6 shows an example with 5 text messages. The first 3 are executed when all of them have been received from the system. The fourth message is executed at once and the last message includes the (x, y, z) coordinates for a new waypoint. The digit 6 in the beginning is an indication for the system that (x, y, z) coordinates follow.

First message: Number of command # message 1 of 3 in total – Arduino is on hold.

Second message: Number of command # message 2 of 3 in total – Arduino is on hold.

Third message: Number of command # message 3 of 3 in total – The commands are executed.

Fourth message: Number of command – the command is executed.

Fifth message: (x, y, z) coordinates – The command is executed.

Fig. 6. Text messages commands sequence.

IV. Connection Times and Field Trials

Field tests were performed and basic control commands were transferred both from a mobile phone and a laptop to the UAV and they were successfully executed, while the UAV was on the ground or in flight. The field trials were repeated on several days and at different times of the day, for evaluating the subsystem under different network load. The subsystem was connected to the same network provider during all tests. The average response time, when the control command was sent as a text message (with GSM) was 2.6 seconds and 0.5 seconds when sent through the internet (with GPRS) using the Mission Planner software.

Specifically, in the text message mode, the initialization time of the Arduino platform is almost 14 seconds, 8 seconds for the GSM shield to connect to the internet and 6 seconds for the Arduino Mega to execute the code. When the initialization phase is finished, the response time depends only on the GSM network delay (2 secs on average) and the processing delay (0.6 secs on average).

In the GPRS mode, the initialization delay is higher, since the UAV configuration is downloaded to the Mission Planner the first time they connect. The download time is 25 seconds on average, the GSM shield needs almost 8 seconds for connecting to the GPRS network, resulting to a total of 33 seconds of initialization time. When the subsystem initialization is over, the response time depends mainly from the network delay, as the processing time is less than 1 msec.

Comparing the two methods, the communication through GPRS using the Mission Planner software is faster, not only because of the network, but because the Arduino Mega just forwards the MAVLink packets without processing them. The subsystem initialization time is higher, but it only happens once, when the subsystem is powered on. The flexibility and the ease of use is also higher, as the Mission Planner offers a user friendly graphical interface, with a variety of configuration options. However, the GPRS or the internet connection is not always available, hence, the communication using GSM and text messages is an efficient alternative in this case.

V. Conclusions

The work presented in this paper opens new perspectives in developing both low-cost and flexible subsystems for transmitting control commands to UAVs, wherever a GSM network is available. The small weight of the equipment makes the implementation possible with even small UAVs. Possible applications include remote UAV access with commands for initializing or adjusting the UAV tasks, even in a different country, or enabling a central base station for controlling multiple remote UAVs, controlling and monitoring their missions more efficiently.

References

[1] "Long Range Tracking Antennas_UNMANNED AERIAL VEHICLES - Micro UAV - Unmanned Aircraft - Marcus UAV", *Marcusuav.com*, 2016. [Online]. Available: http://www.marcusuav.com/largetrackingantenna/.

[2] "UAS Command and Communication - ORBIT", *orbit USA*, 2016. [Online]. Available: http://orbit-cs.com/solutions/land-solutions-uas-command-and-communication/.

[3] "It is now possible to use 3G Connectivity to perform BLOS drone operation! - Delair-Tech", *Delair-Tech*, 2015. [Online]. Available: http://www.delair-tech.com/en/it-is-now-possible-to-use-3g-connectivity-to-perform-blos-drone-operation.

[4] "How it Works", *Skydrone.aero*, 2016. [Online]. Available: http://www.skydrone.aero/fpv/how-it-works-fpv.

[5] L. Zhu, D. Yin, J. Yang and L. Shen, "Research of remote measurement and control technology of UAV based on mobile communication networks," *Information and Automation, 2015 IEEE International Conference on*, Lijiang, 2015, pp. 2517-2522.

[6] A. AbdElHamid, P. Zong and B. Abdelhamid, "Advanced Software Ground Station and UAV Development for NLoS Control Using Mobile Communications", *Discrete Dynamics in Nature and Society*, vol. 2015, pp. 1-20, 2015.

[7] A. AbdElHamid and P. Zong, "Development of UAV Teleoperation Virtual Environment Based-on GSM Networks and Real Weather Effects", *International Journal of Aeronautical and Space Sciences*, vol. 16, no. 3, pp. 463-474, 2015.

[8] S. Qazi, A. S. Siddiqui and A. I. Wagan, "UAV based real time video surveillance over 4G LTE," *2015 International Conference on Open Source Systems & Technologies (ICOSST)*, Lahore, 2015, pp. 141-145.

[9] G. Solidakis, F. Tsokas, M. Batistatos, N. Sagias, G. Tsoulos, D. Zarbouti and G. Athanasiadou, "Arduino (IDE) code for "An Arduino-Based Subsystem for Controlling UAVs Through GSM", submitted to EuCAP 2017 conference", *ResearchGate*, 2016. [Online]. Available: https://doi.org/10.13140/RG.2.2.11583.51363.

[10] "MAVLink Micro Air Vehicle Communication Protocol - QGroundControl GCS", *Qgroundcontrol.org*, 2016. [Online]. Available: http://qgroundcontrol.org/mavlink/start.