

The Bluetooth Smart SensorTag Development Package

Company: "Gh. Asachi" Technical University

Author(s): Dan-Marius Dobrea

NI Product Used: LabWindows CVI

Industry: Aeronautical

Application Area: Guidance, navigation, and control of drones (quadcopters, bi-copters etc.)

The Challenge: Designing a control system for quadcopters is not a simple task. Commonly, the classical approach uses Kalman Filter(s) and PID controllers. In this case, a new challenge emerges when an unknown drone model is used, and this is because of the difficulty of finding the correct parameters for the software modules.

The Solution: By using the advantages provided by LabWindows CVI, two applications were developed and implemented, in order to obtain the best parameters for a drone's controller. The first application tunes the Kalman filter and the second one, monitors, sets and controls the quadcopter.

Introduction

The "Gheorghe Asachi" Technical University of Iasi is among the oldest and well known institutions from the country, having an important tradition in engineering, scientific and cultural education, with a distinguishable presence at the international level. Presently, I am associated professor at this university, activating in the Department of Applied Electronics and Intelligent System, from Faculty of Electronics, Telecommunications and Information Technology. My research interests are in the fields of digital signal processing, intelligent systems, bioinspired robots, human computer interface, cloud computing, and embedded systems.



Figure 1. Different types of drones

Nowadays, the drones' industry has created entirely new markets, with new unprecedented opportunities. Drones are used today in many applications; for instance, they are exploited in: disaster management (search and rescue, firefighting etc.), wildlife, weather and environmental monitoring, thermal infrared for power line, gas and oil pipe surveillance, rural medical services, exploration, law enforcement, military operation and surveillance, television news coverage, aerial imaging, traffic

monitoring, etc. According to a market research investigation, the drones' industry will result in an economic impact of around 82 billion USD between 2015 and 2025, only in USA.

Even if the main concepts, that underlie hovering, taking off, flying and landing, are similar for all different types of drones (see Figure 1), it proves to be very difficult to build a dynamical model for such systems. Mainly, these difficulties are generated by (1) the nonlinear characteristics of these specific types of equipment, as well as, by (2) the constructive diversity of drones. The parameters of the drone's dynamical model are directly correlated with the parameters of the software control system (i.e., the parameters of the Kalman filter and PID - see Figure 2). Because only part of the control parameters can be roughly estimated, a trial and error procedure is required in order to properly tune all control parameters; this tuning is done accordingly with the best dynamical flying characteristic imposed by: (1) the requirements of the owner and (2) the specific domain of use (e.g. acrobatic versus surveillance).

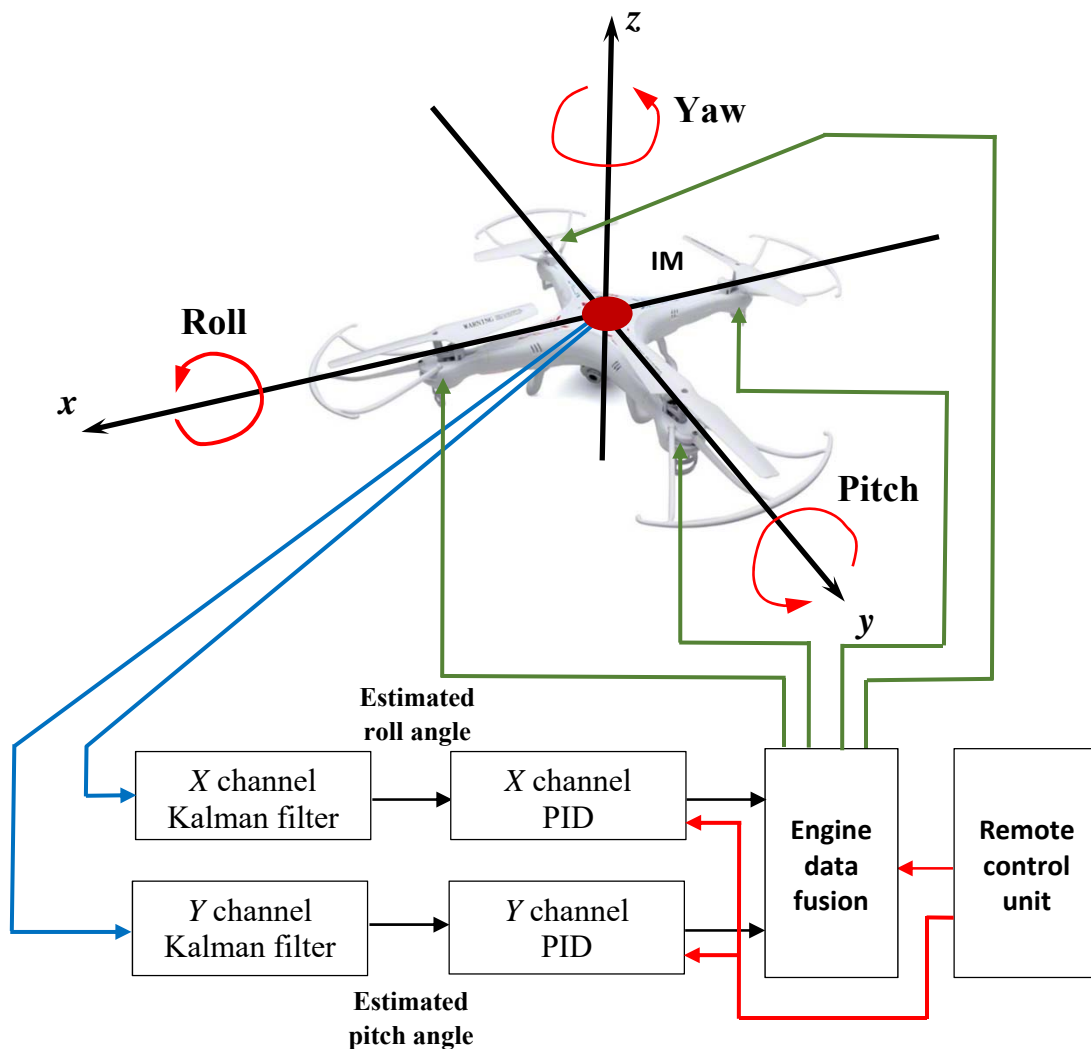


Figure 2. The quadcopter's main software architecture

Application Overview

The system used in this research is a four-rotor Unmanned Air Vehicle (UAV) called quadcopter, **Figure 3**. The quadcopter was developed together with one of my students (Lupuleasa Tiberiu), in the frame of his diploma project. The electronics that control and command the quadcopter consists of: **(1)** a flight stabilizer sensor, which is a nine degree of freedom (gyroscope, accelerometer and compass) Inertial Measurement Unit (IMU) – MPU-9250, **(2)** a 32 bits microcontroller, working at 80 MHz – PIC32MX695F512H and **(3)** a Bluetooth module.

Due to the lack of any prior experience in the control of quadcopter systems, several approaches were used. The identification of an unknown parameter for a quadcopter can be done in two main different ways. In the first methodology, the unknown parameter(s) can be determined by direct computation from the quadcopter model. The second way is by analyzing the flight data.

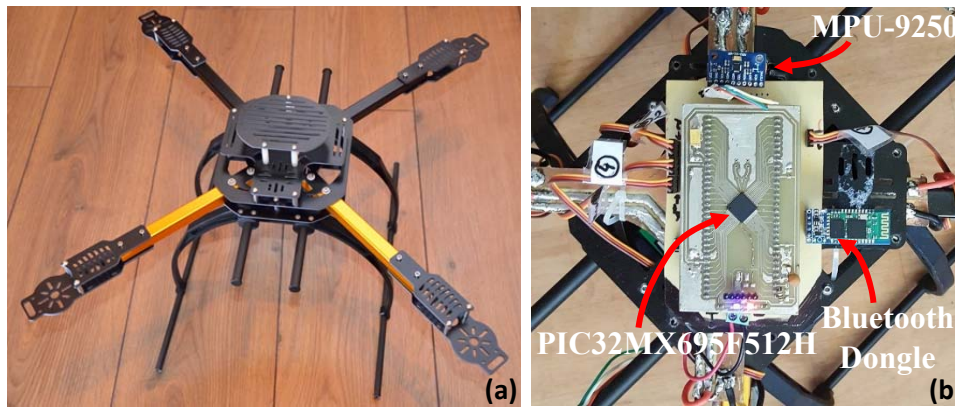


Figure 3. (a) Quadcopter frame, (b) Electrical control unit

Prior to the introduction of the software package, proposed in this paper and developed together with the indicated quadcopter, a model-based design approach for the control system of the quadcopter was followed. The model-based design approach was created using the following triad: SimMechanics, Simulink and Matlab. Unfortunately, it was very difficult and hard to build a realistic 3D CAD model of our quadcopter. As a result of this difficulty, a different approach had to be followed.

In a first step, a specific application was developed to determine and tune the Kalman filter parameters. The Kalman filter was used to calculate a precise angle (i.e., the roll or pitch angle) within a data sensor fusion process, that combined data from both accelerometer sensor and gyroscope sensor, Figure 2.

The Kalman filter is a recursive estimator, having two stages: the prediction and the update stage.

Prediction stage:

- I. Predict the state:

$$\bar{x}_k = A \cdot x_{k-1} + B \cdot u_k + w_k \quad (1)$$

- II. Compute the predicted error:

$$\bar{P}_k = A \cdot P_{k-1} \cdot A^T + Q \quad (2)$$

- III. Sensor prediction:

$$\bar{z}_k = C \cdot \bar{x}_k + v_k \quad (3)$$

Update stage:

IV. Compute the Kalman Gain:

$$K = \frac{\bar{P}_k \cdot H^T}{H \cdot \bar{P}_k \cdot H^T + R} \quad (4)$$

V. Update the estimate with the measurement:

$$x_k = \bar{x}_k + K \cdot (z_k - H \cdot \bar{x}_k) \quad (5)$$

VI. Update the error covariance:

$$P_k = (I - K \cdot H) \cdot \bar{P}_k \quad (6)$$

This paper does not aim to explain how Kalman filter operates. But, the most important step in designing a Kalman filter lies in its tuning – one of its major problems. Tuning is achieved by carefully selecting the Q and R covariance matrices, presented in the equations above. Today, there are some methods to estimate these parameters from data (with some inaccuracy, that are impossible to know); among these methods, the trial and error method stands out as the most extensively used method in “tuning” the filter at the desire performance. Other parameters that are required at the startup of the algorithm - such as the values for the error covariance matrix of the state, P, and the state matrix, \bar{x}_k - are also unknown.

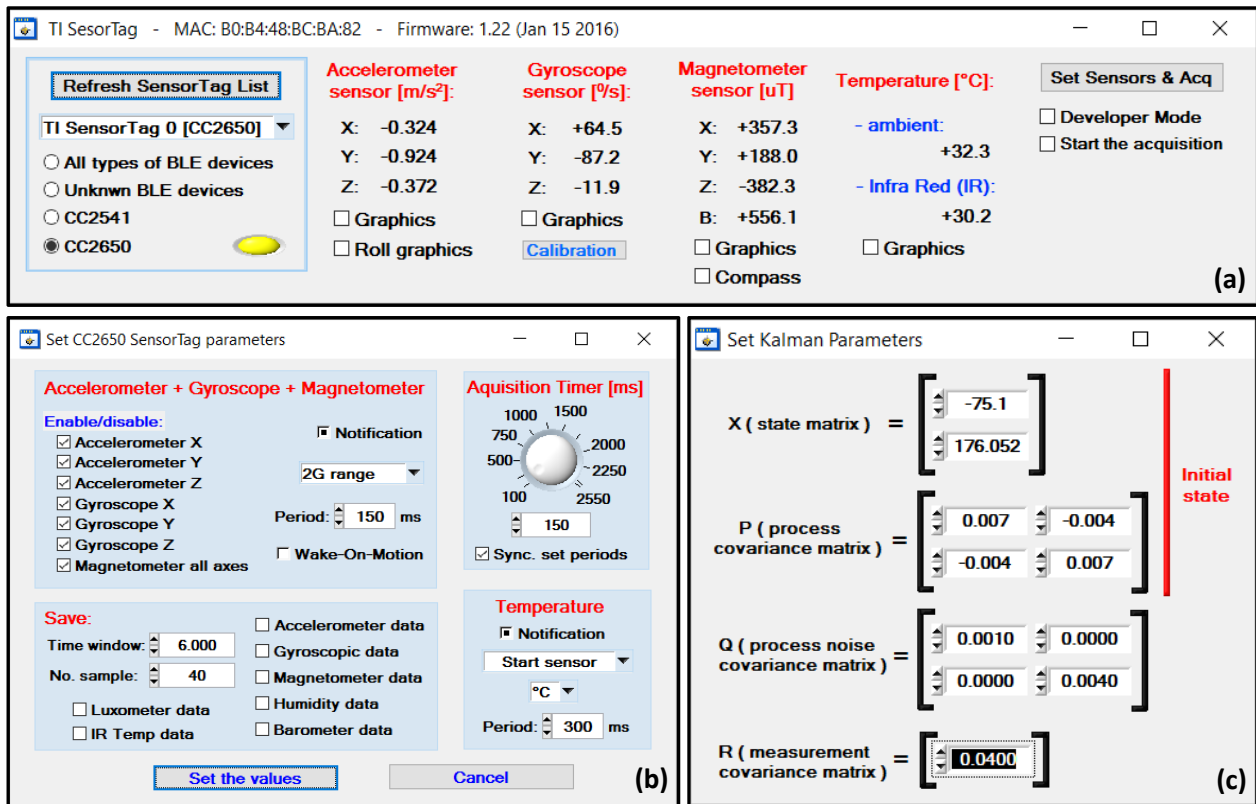


Figure 4. (a) The main UIR, **(b)** UIR for the SensorTag configuration, **(c)** UIR to set the Kalman Filter parameters

Aside from these, during the designing process of our particular drone, we had to take some additional decisions. One of them was tied to the particular configuration assumed for the quadcopter. In its actual configuration, the system, **Figure 3**, uses: (1) two sensors (an accelerometer and a gyroscope) with (2) two

axes each (x and y, **Figure 2**). From here, many approaches could be equally followed to construct the state matrix (\bar{x}_k from equation (1)) – e.g.: **(1)** to combine all sensors and all axes together, **(2)** to run the Kalman filter two times for each axes (as it is presented in **Figure 2**), **(3)** to introduce only the angle and the angle bias on the state matrix or to put also the angular velocity, etc.

In order to choose both the best software architecture and the optimal parameters for the Kalman filter, a special application was developed. This application, which communicates with a CC2650 SensorTag placed on the quadcopter, was designed:

- (1)** to configure (see **Figure 4(a)**) and to get the accelerometer, gyroscopic and magnetic data from the SensorTag, and
- (2)** to present these data in different visual formats and/or to save them to the hard drive – see **Figure 5(a), (b)** and **(c)**.

The SensorTag device works as a GATT server. GATT (Generic Attribute Profile) defines the way that two BLE (Bluetooth Low Energy) devices transfer data back and forth. The CC2650 SensorTag was selected mainly because it incorporates the same MPU-9250 IMU that we also used in our quadcopter control system, see **Figure 3**.

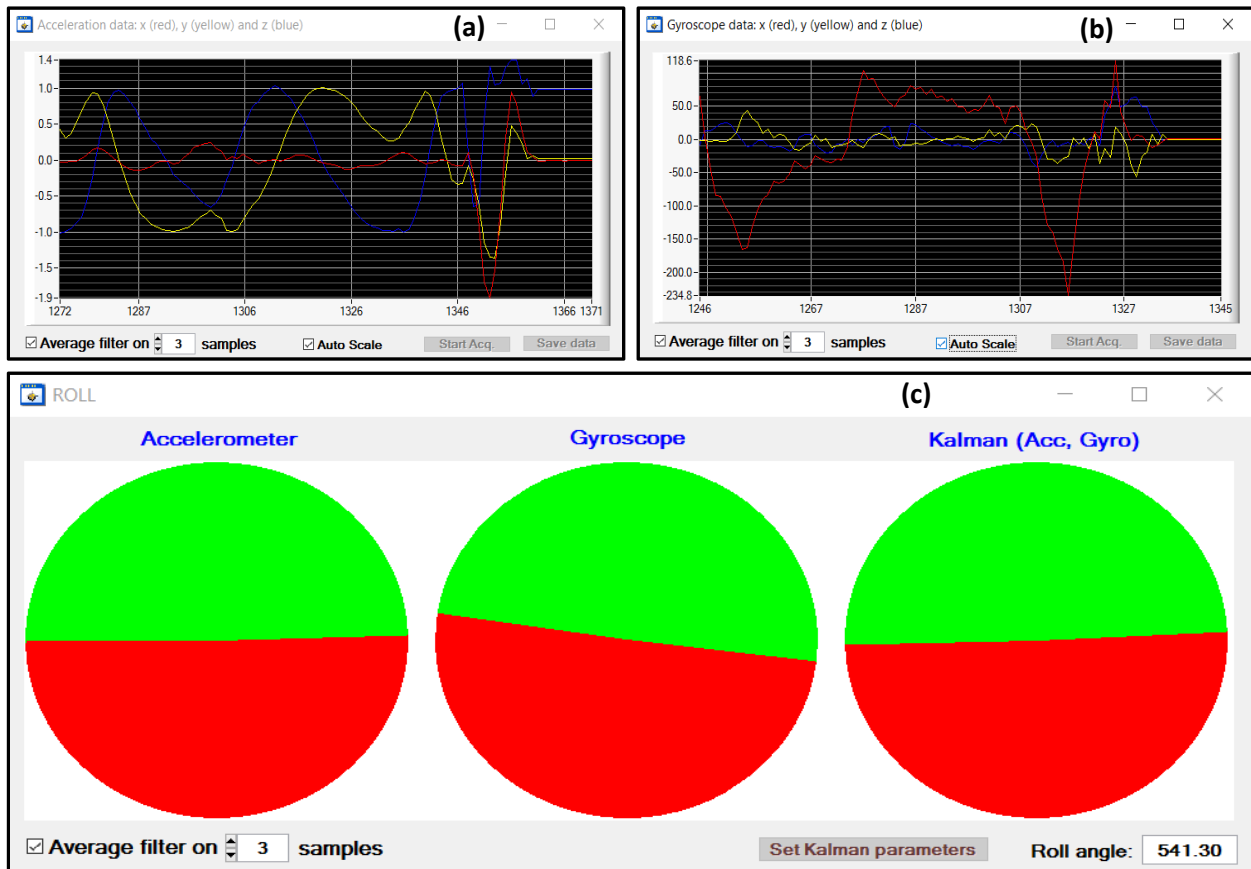


Figure 5. (a) Strip Chart graph of the acceleration data, (b) Strip Chart graph of the gyroscopic data and (c) the roll angle given by the accelerometer, gyroscope and Kalman filter.

Our software package is able to work simultaneously with up to 30 different BLE devices (CC2541 and/or CC2650 SensorTag). Also, a developer mode was implemented. In the developer mode the software provides the user with lots of messages obtained from the communication process with the SensorTag; these messages enable the user to identify the communication/configuration setbacks and some other problems. Moreover, in the developer mode the software gives the possibility to interrogate different types of BLE devices, other than CC2541 and CC2650 SensorTags.

The optimal Kalman filter parameters, obtained with this software package, further feed the Kalman filter implemented with the software programs running inside the PIC32MX695F512H microcontroller.

In the **second step**, another application was developed and used in order to find the optimal parameters for the PID control loop, **Figure 6**. The communication between the quadcopter and the personal computer (that was running this software package) was realized by a Bluetooth protocol that tunnels a standard RS232 serial communication protocol. Another contribution of this second software application is that it plays a relay function. More exactly, it receives commands from a Microsoft XBOX wireless controller (i.e., throttle, forward-backward and left-rights commands) and further sends them to the quadcopter.

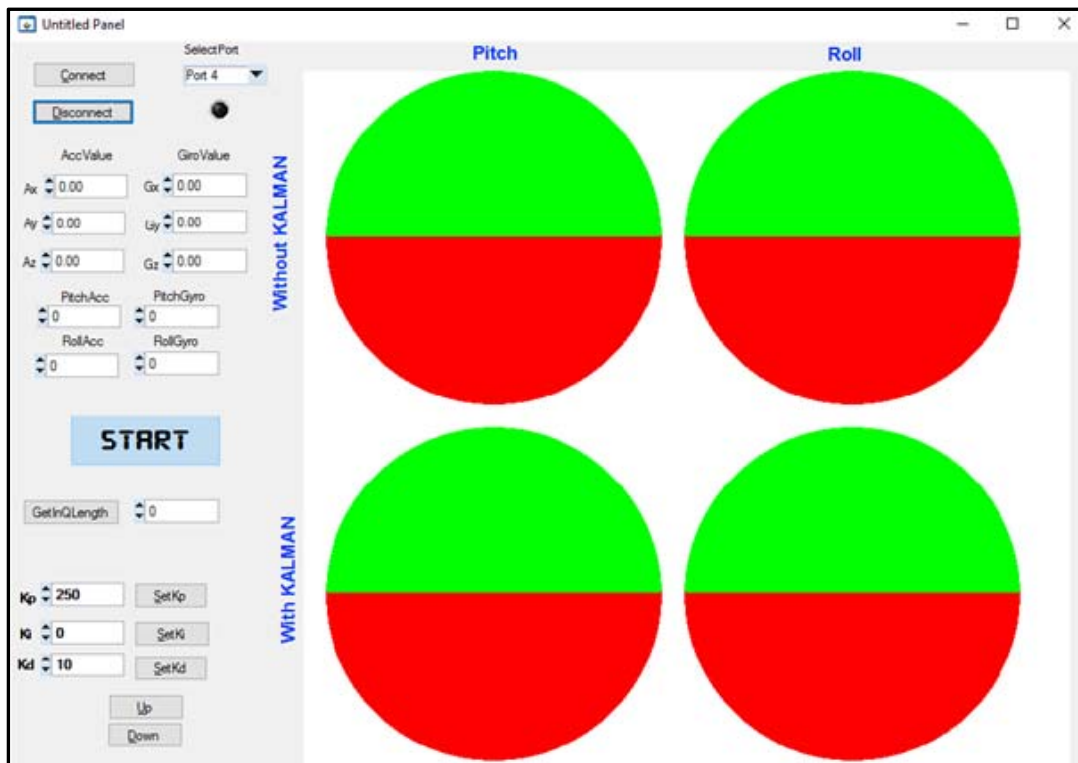


Figure 6. UIR for the drone control system

Both applications were written in C language, using the LabWindows CVI environment. This environment provides a significant advantage in terms of speed and easiness of software development, deployment and testing. A leading benefit that LabWindows CVI brings consists in its powerful algorithms and functions included in the Advanced Data Analysis library – especially, array operations; this kind of operations were widely used in our both developed applications. The ability to interface and easily work with Windows SDK was the second exploited advantage.

At the beginning, the applications was developed using Microsoft Visual Studio environment. Moving from Microsoft Visual Studio to LabWindows CVI environment was a necessity mainly given: (1) by the need of an easy and fast way to communicate and display the information to the user and, also, (b) by the advantages assured by the RS232 library.

Benefits of Your Application

From our experience, with two previous failed attempts to build control units for drones (a bi-copter and a smaller quadcopter), me and two of my former students had to conclude that, without a very flexible set of tools used to tune and test the drone controller unit, it is very difficult and almost impossible to obtain a stabile hovering. For a company, having these two application tools, the development time of a new control unit for this or for any other type of drones will be drastically reduced, with direct benefits in overall production costs. The impact of these applications are, therefore, directly related to a significant cut down of the development time and costs. All drone fields of application and companies emerged in these markets could benefit from the applications presented above.

At this moment, in the Microsoft virtual store six applications are able to interface with Texas Instruments SensorTags (CC2541 and/or CC2650). But unfortunately, none of them was designed to determine the optimum parameters of a Kalman filter. All six applications can only configure a SensorTag and present visually the values acquired from the sensors. Comparing these applications with the BLE SensorTag application, **Figure 4**, we can say, without fail, that it implements entirely new functions (non-existent in all other applications) which, by their practical applicability, define its value and unicity.

Conclusion

In this paper, the main concept of modeling and controlling a quadcopter was presented. One major advantage of this concept is given by the developed software packages that can be applied to a variety of other systems.

With LabWindows CVI, we built two applications. Using these software packages the development of the drone control unit can be done in an efficient, faster and cheaper way, with a small development team.

The LabWindows CVI meets all our expectations. It gave to us all the flexibility and power to implement and test multiple mathematical models. These models are parts of the drone command and control units.

The next logical step in the system development process will be to implement a magnetometer tilt compensation for yaw axis. Moreover, the software package used to tune PID controller can be also improved, especially the microcontroller component, redesigning it as a library for PIC32MX695F512H microcontroller. From this point, several different types of libraries can be developed for different types of microcontrollers – e.g. for several Arduino or ARM family of microcontrollers.

Contact Information

Dan-Marius Dobrea

Address: Faculty of Electronics, Telecommunications and Information Technology, Br. Carol I, Nr. 11A, 700506, Iasi, Romania

Email: mdobrea@etti.tuiasi.ro

LinkedIn: <http://ro.linkedin.com/pub/dan-dobrea/35/aa7/ba3>

Webpage: <http://www.etti.tuiasi.ro:81/cin/Members/Dan/dan.htm>

Facebook: <http://www.facebook.com/mdobrea>