Aeronautical telecommunication simulator

D.K. Nedev, and K.D. Nedev

Abstract—Aeronautical telecommunication simulator is a product specially made for education and training of standard aviation phraseology and flight instruments. It consists of two separate devices: a pilot's radio and navigation panel based on Pilatus PC-9M and an ATC's radio panel. They communicate with each other via FM frequency radio signals. The hardware part of the system is based on Arduino Mega 2560 and Arduino Uno microcontrollers.

Index Terms-aeronautical, telecommunication, phraseology, simulator, Arduino

I. INTRODUCTION

REPORTS claim that air transport is the safeties transport in the world. However, aviation accidents do happen occasionally and harm human life. They are unpredictable, unpreventable and often cause loss of life. A huge percent of all aviation accidents are caused by misunderstanding in communication between pilots and ATCs. Even the worst air disaster of all times - Tenerife airport disaster resulted due to miscommunication. Therefor understanding and practicing aeronautical telecommunications is essential for the flight safety.

The simulator is a collection of hardware and software particularly designed for training aeronautical phraseology between pilots and controllers. It consists of two separate devices: a pilot's radio and navigation panel based on Pilatus PC-9M aircraft and an ATC's radio panel. The unlimited number of panels can be added for each exercise. For example: two pilots and three controllers can practice together in real time. Devices communicate with each other via FM frequency radio signals. In addition, a pilot's Audio Radio Control Display Unit (ARCDU) from PC-9M can be connected to a flight simulator software such as Microsoft Flight Simulator X and X-plane.

II. PILOT'S RADIO AND NAVIGATION PANEL

It is based on Pilatus PC-9m aircraft. It has seven separated boxes. Two of them control VHR radio stations, the other three control navigation stations and there is also one for UHF communication and one transponder identification.

D. NEDEV, works in University of Plovdiv, Technological Faculty, Smolyan, Bulgaria, (e-mail: dim_nedev@abv.bg)

K. NEDEV, works in Graf Ignatievo Air Force Base, Bulgaria (e-mail: koko_nedev@abv.bg)

Each box consists of two frequencies: active and standby, which can be switched by a push button located next to the box. This device can transmit radio signals on two different frequencies if "1-2" mode is turned on. In the same time, it can receive three (two VHF and one UHF) signals. It is also equipped with seven rotary encoders with push buttons for adjusting volume of the incoming signals of each box. With the push button we can choose the active box (the box we are transmitting from).

a)

b)

Fig.1.ARCDU panel from PC-9M & ARCDU simulator

On the right bottom part of the panel there is a frequency dual rotary encoder which is used for adjusting the standby frequency of each box. Using the PGE button main two displayed pages of the device can be switched.
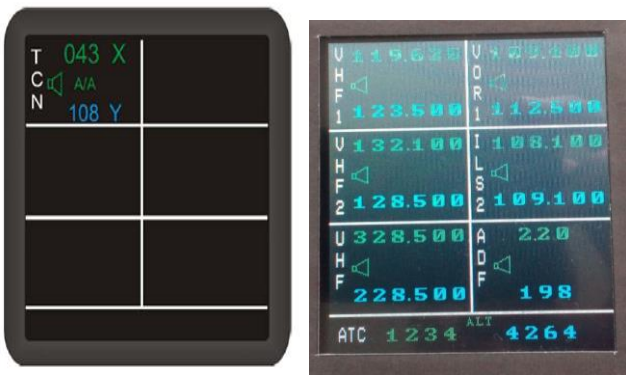
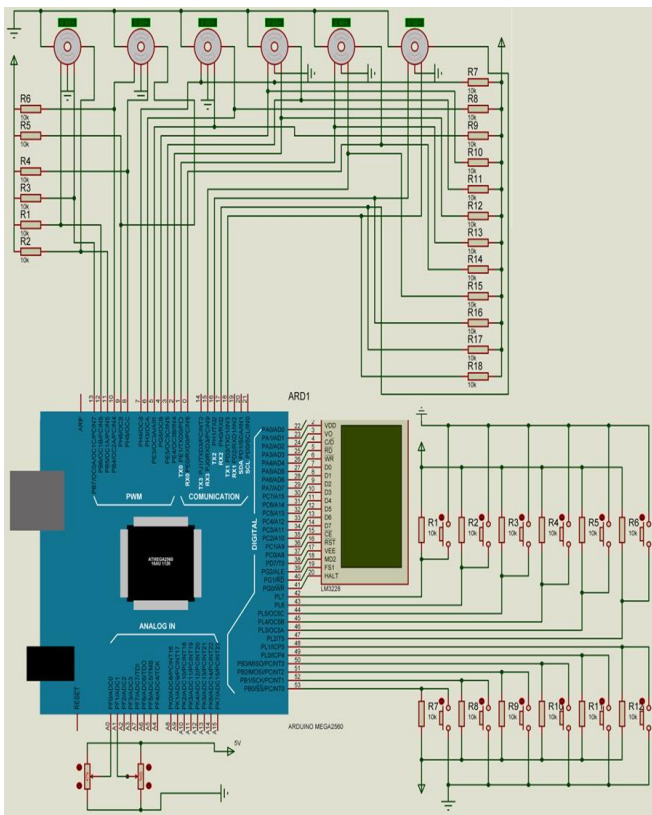Fig. 2. ARCDU page 2 & ARCDU simulator's display

III. HARDWARE

Fig. 3. Wiring diagram

Hardware part is based on Arduino Mega 2560 microcontroller, a 3.2" TFT LCD screen and 12V power supply unit. It also incorporates with 7 encoders, two potentiometers, one switch and sixteen push buttons. Six of the encoders control volume level of each box, seventh is dual and adjusts standby frequency of each box. All hardware parts are connected to microcontroller's digital pins except for the potentiometers which are connected to analog once. The first twenty pins from the Arduino microcontroller are used for the rotary encoders. The next twenty pins link microcontroller with the TFT display. The pins from 42 till 53 are used for controlling the push buttons. All these pins use digital serial communication. There are also two analog potentiometers connected to analog pins A0 and A01. The first one controls the brightness of the TFT display. The other controls the IC.

The radio transmitter and receiver are connected to extension board. The wiring diagram is shown on Fig.3.

IV. SOFTWARE

Software part has been created with Arduino IDE. The code consists of more than two hundred lines. First main function is called "drawHomeScreen ()". It uses UTFT library to visualize home page.

```
//********************************************************
#include <UTFT.h>
// drawHomeScreen - Custom Function
void drawHomeScreen() {
  // Title
  myGLCD.setBackColor(0,0,0); // Sets the background color
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.drawLine(0,92,240,92); // Draw lines
  myGLCD.drawLine(0,93,240,93);
  myGLCD.drawLine(0,189,240,189);
  myGLCD.drawLine(0,190,240,190);
  myGLCD.drawLine(0,282,240,282);
  myGLCD.drawLine(0,283,240,283);
  myGLCD.drawLine(119,0,119,283);
  myGLCD.drawLine(120,0,120,283);

  myGLCD.setFont(Retro8x16); // Sets font to Retro
  myGLCD.print("V",125,5); // Print VOR1
  myGLCD.print("O",125,27);
  myGLCD.print("R",125,49);
  myGLCD.print("1",124,71);

  myGLCD.print("V",5,100); // Print VHF2
  myGLCD.print("H",5,123);
  myGLCD.print("F",5,144);
  myGLCD.print("2",5,166);
```

Fig. 4. Arduino ARCDU code

```
void loop ()
{ // Change active and pasive freq. from Box1
  Box1_buttonstate=digitalRead(45);
  delay(40);
  if (Box1_buttonstate==LOW)
  {
    rotation_big=Box1_active_big;
    rotation_small=Box1_active_small;
    Box1_active_big=Box1_pasive_big;
    Box1_active_small=Box1_pasive_small;
    Box1_pasive_big=rotation_big;
    Box1_pasive_small=rotation_small;
    drawHomeScreen();
  }

  Box1_buttonstate=digitalRead(17);// top left encoder button
  delay(50);
  Serial.println(Box1_encoder);
  if (Box1_buttonstate==LOW)
  { activebox=1;
    clr_screen();
    drawHomeScreen();
    myGLCD.setFont(Retro8x16);
    myGLCD.setColor(15,135, 39);
    myGLCD.drawRect (2, 0, 15, 88);
```

Fig. 5. Arduino ARCDU main function

The code of the program includes more than twenty other custom functions. It also has two standard functions for each Arduino program: void setup () and void loop (). The first one

is executed only once following program startup. It is used to define microcontroller's pins as INPUT or OUTPUT. The second standard function void loop works all the time when the device is turned on. It constantly monitors the status of all encoders, push buttons and potentiometers. The code has modular structure and easily can be modified and updated.

V. ATC RADIO PANEL

It has one 2x8 chars LCD display, two rotary encoders and a switch button. It is also equipped with FM radio receiver and transmitter. It can receive and transmit signal to only one frequency.

a)

b)

Fig. 6. ATC's panel

VI. CONCLUSION

According to our experiments Aeronautical telecommunication simulator is a very useful training system for pilots and ATCs. It can be used from beginner and experienced pilots and controllers. The system is part of a new aviation education system used in Bulgarian Air Force, which is going to be announced in February 2017.

REFERENCES

[1] Krushkov Hr., C++ programming, Plovdiv, 2006

[2] Kenarov N., PIC microcontrollers, Varna, 2006

[3] Kenarov N., Microcontrollers in exercises, Varna, 2006

[4] http://howtomechatronics.com/

[5] www.arduino.cc

[6] Jeremy Blum, Exploring Arduino, 2016

[7] Kraig Mitzner, Complete PCB Design Using OrCAD

[8] James M. Kirkpatrick, Electronic Drafting and PCB, Second edition

BIOGRAPHIES

DIMITAR K. NEDEV was born in Smolyan, Bulgaria. He received the B.S. and M.S degrees in Microelectronics from "Angel Kanchev" University of Ruse, Ruse, Bulgaria. Since 2015, he is a lecturer in "Paisii Hilendarski" University of Plovdiv, Bulgaria. Her research interests are in electronics and particularly in designing PCBs.

KOSTADIN D. NEDEV was born in Smolyan, Bulgaria. He received the B.S. degree in Aviation from National Military University, Aviation Faculty, Dolna Mitropolia, Bulgaria. He has passed also ATPL course in Polish Air Force Academy, Deblin, Poland. Since 2016 he works as a pilot in Bulgarian Air Force.