Article **ECORFAN-Mexico Journal**
June 2018 Vol.9 No.20 66-70

Application to control RC devices in the frequency of 49 MHz using C# .Net and Arduino

Aplicación para controlar dispositivos RC en la frecuencia de 49 MHz usando C # .Net y Arduino

ABRIL-GARCIA- José[1]†*, MEZA-IBARRA, Iván[1], ALCÁNTAR-MARTÍNEZ, Adelina[1], GARCÍA-JUÁREZ- Alejandro[1]

[1]*Universidad Tecnológica de Hermosillo, Blvd. de Los Seris final sur s/n., Hermosillo, Sonora, México*
[2]*Universidad de Sonora, Departamento de Investigación en Física*

ID 1°Author: *José Humberto Abril-García* / **ORC ID:** 0000-0003-3494-6817, **Researcher ID Thomson:** F-4252-2018, **arXiv ID:** jhabril, **CVU CONACYT ID:** 204935

ID 1° Co-author: *Iván Dostoyewski Meza-Ibarra* /**ORC ID:** 0000-0001-6139-032X, **Researcher ID Thomson:** F-3550-2018**, arXiv ID:** imeza, **CVU CONACYT-ID:** 769494.

ID 2° Co-author: *Adelina del Carmen Violeta Alcántar-Martínez* / **ORC ID:** 0000-0003-2715-9209, **Researcher ID Thomson:** F-6771-2018, **CVU CONACYT-ID:** 640868

ID 3° Co-author: *Alejandro García-Juárez* / **ORC ID:** 0000-0002-7625-3093, **Research ID Thomson:** F-6868-2018, **arXiv ID:** agarcia72, **CVU CONACYT-ID:** 201129

Received January 25, 2018; Accepted June 12, 2018

Abstract

The interest for using and adapting technology is increasing today, primarily for the use of control devices of different ways or techniques. A practical guide of an operable system composed by a computer application and electronic circuits is designed to control a Radio-Controlled device (toy car) in the frequency of 49 MHz and using a C# .NET GUI to simulate the control on the computer screen which communicates with an Arduino Board to control the device. This kind of application base, which involves electronic circuits and software, can be used as methodic practice with students to develop better programming and electronic skills.

RC, GUI, Arduino, PCB

Resumen

El interés por usar y adaptar tecnologías está aumentando actualmente, principalmente el uso de controles para diferentes dispositivos. En este trabajo se presenta una guía práctica de un sistema compuesto por una aplicación de computadora y un circuito electrónico para operar un dispositivo controlado por radio (carro de juguete) en la frecuencia de 49MHz, utilizando una GUI en C# .NET para simular el control en una computadora, la cual se comunica con una placa Arduino para controlar el dispositivo. Este tipo de aplicación, que incluye circuitos electrónicos y software, puede ser usada como una práctica metódica con estudiantes para desarrollar mejores programas más complejos y robustos, así como habilidades en el área de la electrónica.

RC, GUI, Arduino, PCB

Citación: ABRIL-GARCIA- José, MEZA-IBARRA, Iván, ALCÁNTAR-MARTÍNEZ, Adelina, GARCÍA-JUÁREZ- Alejandro. Application to control RC devices in the frequency of 49 MHz using C# .Net and Arduino. ECORFAN Journal-Mexico 2018, 9-20: 66-70

* Correspondence to Author (email: abril@uthermosillo.edu.mx)
† Researcher contributing first author.

I. Introduction

In many systems, the capability of Radio Control (RC) transmission is essential; Control systems, Internet of things (IoT), medical and mobile applications, are examples for these communications that could be either wide or local access. Other proposals are (Zhu, 2011). It is essential to know the basic structure of these systems to develop more complex and robust applications that increase the use of RC. An experimental application that consists of a C# .NET Graphic User Interface (GUI) that sends and receives signals across a Universal Serial Bus (USB) port from an electronic system integrated by Arduino, a Printed Circuit Board (PCB) and RF control, is proposed to allow an RF transmission to control a small device, in this particular case a toy car. Figure 1 shows the set up used to develop the application.

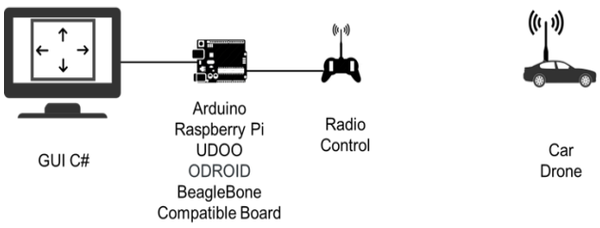

Figure 1 Develop process of the application

At first, a C# .Net GUI is designed to simulate the functionality of the real control operation and then an Arduino board is programmed to receive the signal from a computer using an USB port. Then the GUI was tested to simulate the control, with a four LEDs array on a proto-board that represents the control buttons (forward, backward, left and right). A re-engineering in order to know the way how the controls work was made, the process consisted on analyzing the control's internal functionality.

Then an integration of the control with Arduino board was made; finally, a PCB circuit to assemble all the components in one single block was designed. The main contribution of this paper is to develop a general scheme for RC communications to control electrical devices, additionally the experiment allows having the opportunity to understand concepts related with RC transmissions, programming and electronics, besides the experiment is quite suitable for different courses about this kind of topics.

The rest of the paper is organized as follows: Section 2 describes the application and results; Section 3 gives our conclusion to this work. Furthermore, analysis and future use for this application are provided.

II. Development

GUI and program computer development to simulate RC control

Once designed the GUI on Windows Form as in (Brown, 2006), the necessary code to simulate the actions of the buttons at the RC, was made on C# .NET (Appendix A). When a user presses a button, the key address on the keyboard is detected. If the user tries to press up and down or left and right keys at same time the application will only respond to the first pressed key and will block the second pressed key validating the integrity and functionality of the application. The GUI will show which key is being pressed as shown in Figure 2, by the changing the color of the pressed arrows on the screen.

Figure 2 GUI of application, the left button key is being pressed

Programming Arduino to communicate with the RC car

To interact with the application and RC control, an additional circuit is needed to send digital electrical pulses, so it can control the RC car; a micro Arduino is used (Z. Wang, 2014). In this case it was needed to send signals indicating that "0" is a low state (OFF, LOW) and "1" is a high state (ON, HIGH) (Appendix B). Then the GUI was programmed to communicate with Arduino through the USB port to perform an integration of both technologies.

ISSN-Print: 2007-1582- ISSN-On line: 2007-3682

When the GUI and Arduino were integrated, a test was made by sending electrical pulses to Arduino to interact with the GUI pulsing buttons; to emulate the RC, an array of four LEDs on a proto-board were adapted, in order to simulate the key pressed on the GUI thus the expected result was obtained. This is shown in Figure 3.

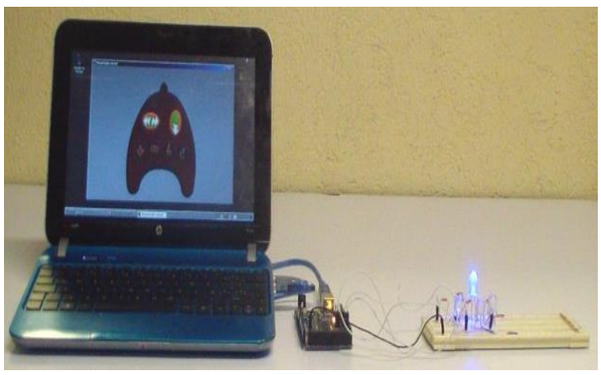

Figure 3 The GUI program simulating the RC car control.

Understanding the RC car

It was necessary to know the internal connections and functionality of the RC car in order to manipulate it from the Arduino. The RC car was disassembled, and then electrical probes with a power supply were applied directly on its electrical inputs. Notice that the wireless signal of RC is at 49 MHz Another fact to considerate is a voltage difference between Arduino and the RC car (5v and 9v), for this reason a circuit formed by opto-couplers was adapted to couple the voltage difference of the devices, in order to protect them as in (A. Thaduri, 2011).

The opto-couplers circuit was developed in house at all (Brooks, 2003) and (Rossano, 2013), it means that it was designed according to the voltage difference, and then simulated and finally printed on a PCB connected between Arduino and the RC car. Figure 4 shows the simulation made at Proteus software, once the circuit functioned correctly, finally the PCB was created as shown on Figure 5.

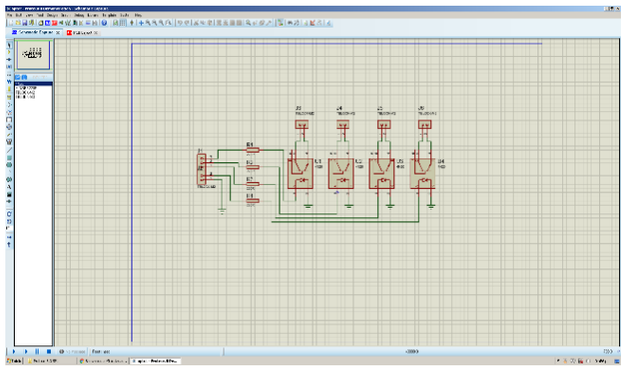

Figure 4 Opto-couplers circuit simulated at Proteus

Figure 5 PCB of the opto-couplers circuit.

The PCB was tested in two different ways:

1. Simulation using Proteus software, functionality and design.
2. Once a correct response from Proteus was obtained physical voltage checking in the four inputs and outputs were made.

The components used on the PCB were:

– One phenolic board 5 cm x 10 cm.
– Seven 2-pin Screw Terminal Block Connector 5.08 mm Pitch Panel PCB.
– Four 100 Ohm Resistor.
– Four optocoupler 4N26

Integration of all components

Once all components were developed and tested, the integration was set as follows:

– GUI and code on C#
– Arduino code
– PCB opto-couplers
– RC car understanding

ISSN-Print: 2007-1582- ISSN-On line: 2007-3682

After the integration was made, the system was tested pressing the keys on a computer associated to the GUI buttons, signals were sent to the Arduino-PCB opto-couplers circuit which in turn sent the signals to the RC car, and then the toy car is manipulated. The complete system is shown on Figure 6.

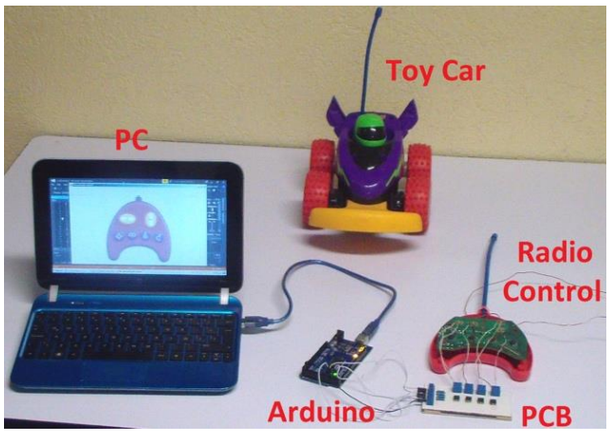

Figure 6 Complete integrated system

III. Conclusions and results

A system composed by software developed on C# .NET that consists of a GUI that simulates a RC car control. A code loaded on Arduino and PCB opto-couplers to send signals to the RC car was designed, developed, studied and analyzed to obtain knowledge about how a device that is controlled by a remote via can be manipulated from a computer, cellphone, tablet or another mobile device. Now a practical guide is available to generate different applications on mechatronics, software development and other areas, for example, manipulation of devices through Internet, using a mobile device. This paper could be used as a guide to teach about software development and electronic topics.

As a result, we develop a methodology too control any radio-controlled device by a GUI on a high programming language.

Appendix A: C# Code

```
public partial class Form1 : Form
{
        private SerialPort port;
       public Form1()
    {
                InitializeComponent();
                    openPort();
    }
       private void Form1_KeyDown(object sender, KeyEventArgs e) {
if (e.KeyValue == 38 & pB2.Visible == false)
        {
         pB3.Visible = true;
                port.WriteLine("A"); // UP
        }
if (e.KeyValue == 40 & pB3.Visible == false)
        {
                        pB2.Visible = true;
                        port.WriteLine("D"); // DOWN
        }
if (e.KeyValue == 39 & pB4.Visible == false)
        {
                        pB5.Visible = true;
                        port.WriteLine("B"); // RIGHT
        }
if (e.KeyValue == 37 & pB5.Visible == false)
        {
                        pB4.Visible = true;
                        port.WriteLine("C"); // LEFT
        }
      }
       private void Form1_KeyUp(object sender, KeyEventArgs e){
                if (e.KeyValue == 38) { pB3.Visible = false; port.WriteLine("a"); }
        if (e.KeyValue == 40) { pB2.Visible = false; port.WriteLine("d"); }
        if (e.KeyValue == 39) { pB5.Visible = false; port.WriteLine("b"); }
        if (e.KeyValue == 37) { pB4.Visible = false; port.WriteLine("c"); }
      }
      private void openPort()
      {
         port = new SerialPort("COM3", 9600);
         port.Open();
              for (int i = 97; i < 100; i++)
                        port.WriteLine((char)i + "");
      }
   }
```

Appendix B: Arduino Code

```
 void setup() {
  Serial.begin(9600);
 pinMode(22, OUTPUT); // UP
 pinMode(26, OUTPUT); // RIGTH
 pinMode(30, OUTPUT); // LEFT
 pinMode(32, OUTPUT); // DOWN
}
void loop() {
 switch (Serial.read())
 {
   case 'A': digitalWrite(22, HIGH); break;
   case 'a': digitalWrite(22, LOW);  break;
   case 'B': digitalWrite(26, HIGH); break;
   case 'b': digitalWrite(26, LOW);  break;
   case 'C': digitalWrite(30, HIGH); break;
   case 'c': digitalWrite(30, LOW);  break;
   case 'D': digitalWrite(32, HIGH); break;
   case 'd': digitalWrite(32, LOW);  break;
  }
}
```

ISSN-Print: 2007-1582- ISSN-On line: 2007-3682

References

A. Thaduri, A. V. (November de 2011). Reliability prediction of opto-couplers for the safety of digital instrumentation. Quality and Reliability, 491-495.

Brooks, D. (2003). Signal Integrity Issues and Printed Circuit Board Design. (P. Hall, Ed.) United States of America.

Brown, E. (2006). Forms Programing with C#. (M. P. Co., Ed.) United States of America.

Rossano, V. (2013). Proteus VSM: Introducción a la simulación en Proteus. (C. A. Corp, Ed.) Estados Unidos de América.

Z. Wang, E. G. (November de 2014). Design of an arduino-based smart car. SoC Desgin Conference, 175-176.

Zhu, W. Q. (2011). Remote control of model car by using mobile technology. Commution Software And Networks, 188-191.

ISSN-Print: 2007-1582- ISSN-On line: 2007-3682