

Monitoreo de control Acteck AGJ-3350 utilizando C# WinForms y Windows presentation foundation

ABRIL-GARCÍA, José Humberto†, MEZA-IBARRA, Iván Dostoyewski y GARCÍA-JUÁREZ, Alejandro

Universidad Tecnologica de Hermosillo, Parque Industrial, 83118 Hermosillo, Sonora

Recibido 3 de Octubre, 2017; Aceptado 8 de Diciembre, 2017

Resumen

Proponemos el desarrollo de una aplicación que muestra en pantalla una Interfaz Gráfica de usuario donde se visualiza un control Acteck AGJ-3350 y se despliega en pantalla las acciones ejecutadas sobre el control utilizando C# WinForms y Windows Presentation Foundation las cuales pueden ser usadas para desarrollar aplicaciones más complejas, que integren diferentes tecnologías, la metodología utilizada para el prototipo fue el desarrollo rápido de aplicaciones, obtenido un resultado de dos aplicaciones, una utilizando WinForm y la otra en Windows Presentation Foundation, las cuales pueden ser empleadas en la integración de proyectos más complejos o como material de clase.

GUI, WinForms, WPA, AForge.NET, Control

Abstract

We propose the development of an application that shows a Graphical User Interface on the screen where an Acteck AGJ-3350 control is displayed and the actions executed on the control are displayed on screen using C # WinForms and Windows Presentation Foundation which can be used to develop applications More complex, integrating different technologies, the methodology used was the development paradigm used was the construction of prototypes, the result was two applications one using WinForm and another in Windows Presentation Foundation, which can be used in the integration of projects more complex or as class material.

GUI, WinForms, WPA, AForge.NET, Control

Citación: ABRIL-GARCÍA, José Humberto, MEZA-IBARRA, Iván Dostoyewski y GARCÍA-JUÁREZ, Alejandro. Monitoreo de control Acteck AGJ-3350 utilizando C# WinForms y Windows presentation foundation. Revista de Innovación Sistemática 2017. 1-4:1-6

† Investigador contribuyendo como primer autor.

Introducción

Actualmente el diseño, desarrollo y comercio de videojuegos es una de las industrias más rentables, inclusive sobre la televisión y cine (Market Brief — Year in Review 2016, 2017), por esta razón consideramos que es importante para los estudiantes de tecnologías de la información o carreras afines conocer las bases de la programación de videojuegos, por lo que proponemos el desarrollo de una aplicación que muestra en pantalla una Interfaz Gráfica de usuario (GUI) donde se visualiza un control Acteck AGJ-3350 (Acteck - AGJ-3350, 2017) y se muestra en pantalla las acciones ejecutadas sobre el control utilizando C# WinForms y Windows Presentation Foundation (WPF) (Microsoft, 2017), las cuales pueden ser usadas para desarrollar aplicaciones más complejas, que integren diferentes tecnologías y que pueda ser implementada en un clase de programación como ejemplo práctico (proponemos la asignatura de Programación de Aplicaciones de la Ingeniería en Tecnologías de la Información), en la figura 1 mostramos el diagrama general del proyecto.

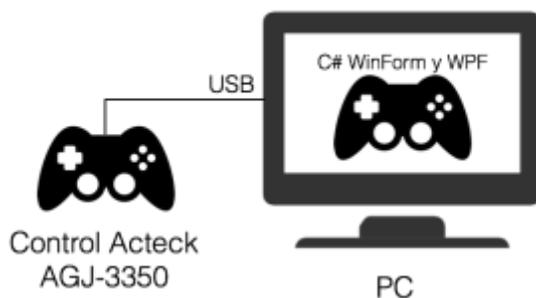


Figura 1 Diagrama general

Descripción de las herramientas utilizadas.

Para el desarrollo de la aplicación se utilizaron las siguientes herramientas:

Visual Studio 2017 Community (VS 2017C) es un IDE (Microsoft, 2017) para la creación de aplicaciones no empresariales para Android, iOS, Windows, Web y la nube.

VS 2017C presenta muchas mejoras en comparación con versiones anteriores entre las cuales podemos mencionar:

- Se ha optimizado para reducir el tiempo de inicio y el tiempo de carga de la solución. El primer lanzamiento de Visual Studio es al menos un 50% más rápido.
- Supervisará ahora el rendimiento de las extensiones que afecten al arranque, la carga de la solución o la edición.
- La extensión GitHub está disponible para instalación.
- Reload All Projects ha sido reemplazado por Reload Solution para soportar un mejor rendimiento al cambiar ramas externas a VS.
- Se instala más rápido, con menos impacto en el sistema, y se desinstala de manera limpia.
- Facilita la selección e instalación de las funciones que necesitas.

Adicionalmente el tipo de licenciamiento, la facilidad de uso y el contacto previo de la mayoría de los estudiantes hacen de esta herramienta la más adecuada para el desarrollo del proyecto.

Otra herramienta utilizada en el desarrollo del proyecto es AForge.NET (AForge.NET © 2008-2012, 2008), un framework open source escrito en C#, diseñado para desarrolladores e investigadores en los campos de Visión por Computadora e Inteligencia Artificial, procesamiento de imágenes, redes neuronales, algoritmos genéticos, lógica difusa, aprendizaje de máquinas, robótica, entre otras.

El framework está compuesto por las siguientes librerías:

- AForge.Imaging: biblioteca con rutinas de procesamiento de imágenes y filtros.
- AForge.Vision: biblioteca de visión por ordenador.
- AForge.Video: conjunto de bibliotecas para procesamiento de video.
- AForge.Neuro: biblioteca de computación de redes neuronales.
- AForge.Genetic: biblioteca de programación de evolución.
- AForge.Fuzzy: biblioteca de cálculos difusos.
- AForge.Robotics: biblioteca que proporciona soporte para algunos kits de robótica.
- AForge.MachineLearning: biblioteca de aprendizaje de máquinas.

El framework proporciona diferentes bibliotecas y sus fuentes, aplicaciones de ejemplo, que demuestran el uso del mismo, y con documentación, en formato de Ayuda HTML. La documentación también está disponible en línea.

Debido a las características del framework, fue seleccionado como soporte a C# para identificar el control e interactuar con él.

Metodología de Investigación

A continuación, listamos las herramientas utilizadas para el desarrollo del proyecto.

Hardware:

- Computadora Pentium(R) Dual-Core CPU @ 3.00 GHz 3.00 GHz, RAM 4.00 GB
- Control marca Acteck modelo AGJ-3350

Software:

- Windows 7 Home Basic 32 bits
- Microsoft Visual Studio Community 2017 VisualStudio/15.0.0+26228.10
- AForge.NET Framework 2.2.5

El primer paso en el desarrollo del Proyecto es la instalación del software y hardware necesarios, en el siguiente orden

1. Visual Studio Community Edition 2017, debido a las nuevas características es necesario realizar una instalación personalizada indicando que se desarrollaran aplicaciones WinForms y WPF. En la figura 2 vemos el instalador de VS.

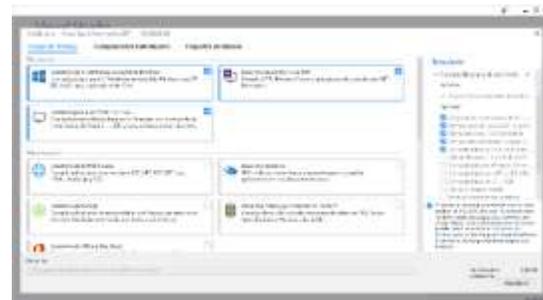


Figura 2 Instalador de VS 2017

2. AForge.NET (figura 3) en este caso es una instalación bastante sencilla donde solo se debe tener cuidado en el directorio destino.

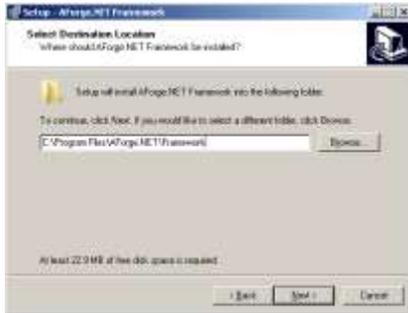


Figura 3 Instalador de AForge.NET

3. Instalar el control, una vez conectado será necesario el uso drivers, en caso de no contar con ellos se pueden descargar de la página oficial del fabricante (Acteck, 2017).

El paradigma de desarrollo utilizado fue el de construcción de prototipos (Pressman, 2005), el cual define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

En otros casos, el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la iteración humano-máquina. En estas, y en muchas otras situaciones, el modelo de construcción de prototipo puede ofrecer el mejor enfoque.

A pesar de que la construcción de prototipos se puede utilizar como un modelo de proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquier otro modelo de desarrollo.

Sin importar la forma en que este se aplique, el modelo de construcción de prototipos ayuda a los desarrolladores y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

De manera ideal, el prototipo debería servir como un mecanismo para identificar los requisitos del software. Si se construye un prototipo de trabajo, el desarrollador intenta emplear los fragmentos del programa ya existentes o aplica herramientas (como generadores de informes, administradores de ventanas, etcétera) que permiten producir programas de trabajo con rapidez.

A continuación, describimos los prototipos generados en cada iteración hasta lograr la versión final.

1. En el primer prototipo se logra la identificación y conexión con el control.
2. En la segunda iteración se logra identificar los botones presionados.
3. En la tercera iteración se realizó el diseño de las interfaces.
4. En la cuarta y última iteración asigno cada acción del control a la imagen correspondiente del diseño.

Se decidió realizar la aplicación en las dos tecnologías debido a la recomendación de Microsoft sobre las mejoras que presentan WPF y el posible desuso de WinForms (Microsoft, 2017).

Una vez instalado todo el software y hardware se procede al desarrollo de las aplicaciones, en el primer caso iniciamos un nuevo proyecto, se programaron todos los métodos necesarios para detectar el dispositivo, identificar los elementos presionados y mostrarlos en pantalla en un objeto de tipo WinForms, en este caso se realizó el diseño de la aplicación (figura 4).

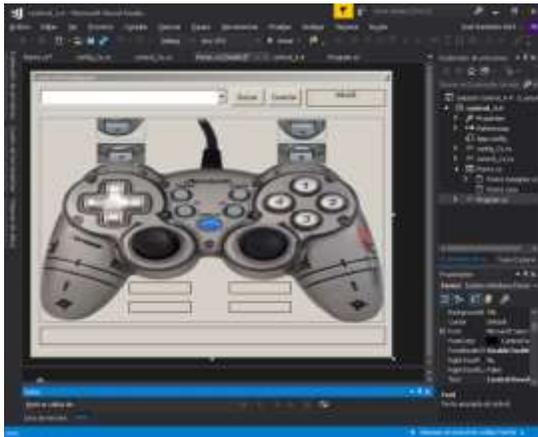


Figura 4 Diseño de la aplicación en WinForms

El mismo proceso se repite para la aplicación utilizando WPF en la figura 5 vemos el diseño de la aplicación.

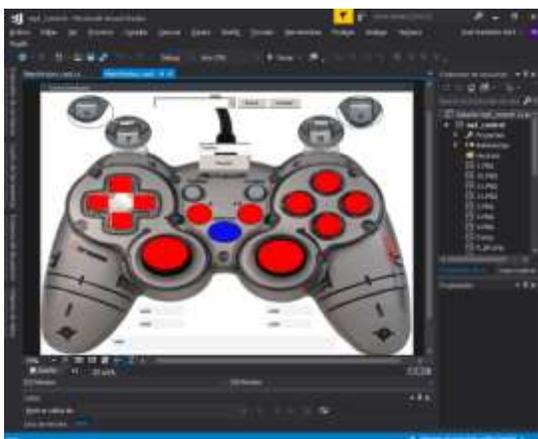


Figura 5 Diseño de la aplicación en WPF

Resultados y conclusiones

En la figura 6 mostramos la aplicación en ejecución usando WPF y WinForms, obteniéndose dos prototipos que pueden ser utilizados de manera didáctica o como base para el desarrollo de aplicaciones más complejas. Como trabajo futuro proponemos desarrollar prototipos de control electrónico donde se utilicen las aplicaciones desarrolladas.



Figura 6 Aplicación en WPF y WinForms

Agradecimientos

Queremos agradecer a la empresa Developnet por las facilidades y equipo otorgado para la realización de este proyecto.

Referencias

Acteck - AGJ-3350. (8 de Mayo de 2017). Obtenido de Acteck: <http://acteck.com.mx/producto.php?id=UEGA-027&fam=PLAY%20IT>

Acteck. (2017). *Acteck Soporte*. Obtenido de Acteck Soporte: <http://acteck.com.mx/rma/>

AForge.NET © 2008-2012. (2008). *AForge.NET Framework*. Recuperado el 8 de Mayo de 2017, de AForge.NET: <http://www.aforgenet.com>

Market Brief — Year in Review 2016. (8 de Mayo de 2017). Obtenido de SUPERDATA GAMES & INTERACTIVE MEDIA INTELLIGENCE: <https://www.superdataresearch.com/market-data/market-brief-year-in-review/>

Microsoft. (2017). *Novedades de Visual Studio 2017*. Recuperado el 8 de Mayo de 2017, de Microsoft Visual Studio: <https://www.visualstudio.com/es/vs/whatsnew/>

Microsoft. (2017). *Windows Forms and WPF Applications*. Recuperado el 8 de Mayo de 2017, de Microsoft Developer Network: [https://msdn.microsoft.com/en-us/library/ff664430\(v=pandp.50\).aspx](https://msdn.microsoft.com/en-us/library/ff664430(v=pandp.50).aspx)

Microsoft. (2017). *Windows Presentation Foundation*. Recuperado el 8 de Mayo de 2017, de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/ms754130\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ms754130(v=vs.110).aspx)

Pressman, R. S. (2005). *Ingeniería del Software Un enfoque practico*. McGraw-Hill.

Anexos

Código de la aplicación WinForms

```
namespace control_4_4
{
    public partial class Form1 : Form
    {
        control_Ce cn;
        ComboBox cole;

        public Form1()
        {
            InitializeComponent();
            cn = new control_Ce();
            buscar_controles();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (cn.Conectar()) { button1.Text = "Conectado"; timer1.Start(); }
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            if (cn.get_palanca_derecha_ar_ab == 400)
            {
                timer1.Stop();
                buscar_controles();
                comboBox1.Items.Clear();
                comboBox1.Text = "no hay";
                return;
            }

            if (label3.Text == "-0.007827878" || label2.Text == "-0.007827878")
            {
                img_image = Image.FromFile("conexo_gla.png");
            }
            else
            {
                img_image = Image.FromFile("conexo_wpf.png");
            }

            if (label4.Text == "-1")
            {
                if (label2.Text == "-0.007827878")
                {
                    far_image = Image.FromFile("f1.png");
                }
            }
            else
            {
                far_image = Image.FromFile("f12.png");
                if (label4.Text == "1")
                {
                    if (label3.Text == "-0.007827878")
                    {
                        far_image = Image.FromFile("f14.png");
                    }
                }
            }
        }
    }
}
```

```
fab_image = Image.FromFile("f14.png");
}
else
{
    far_image = Image.FromFile("f12.png");
    fab_image = Image.FromFile("f14.png");
}
if (label3.Text == "-1")
{
    if (label2.Text == "-0.007827878")
    {
        far_image = Image.FromFile("f14.png");
    }
}
else
{
    if (label3.Text == "1")
    {
        if (label2.Text == "-0.007827878")
        {
            fde_image = Image.FromFile("f13.png");
        }
    }
    else
    {
        far_image = Image.FromFile("f13.png");
        fde_image = Image.FromFile("f1.png");
    }
}

if (cn.boton) { pt.visible = true; }
else { pt.visible = false; }
// Se repite para cada botón hasta el 12

private void buscar_controles()
{
    int cont = 0;
    comboBox1.Items.Clear();
    if (cn.get_dispositivos.Count > 0)
    {
        foreach (var item in cn.get_dispositivos)
        {
            cont++;
            if (cont == 1)
            {
                comboBox1.Text = item.name;
            }
            else
            {
                comboBox1.Items.Add(item.name);
            }
        }
    }
}

private void buscar_controles()
{
    int cont = 0;
    comboBox1.Items.Clear();
    if (cn.get_dispositivos.Count > 0)
    {
        foreach (var item in cn.get_dispositivos)
        {
            cont++;
            if (cont == 1)
            {
                comboBox1.Text = item.name;
            }
        }
    }
}

private void buscar_controles()
{
    comboBox1.Items.Clear();
    return;
}

if (lbl_izrab.Content.ToString() != "-0.007827878" || lbl_dside.Content.ToString() != "-0.007827878")
{
    img.visibility = Visibility.Visible;
}
else
{
    img.visibility = Visibility.Hidden;
}

if (cn.boton) { pt.visibility = Visibility.Visible; }
else { pt.visibility = Visibility.Hidden; }

if (lbl_izrab.Content.ToString() == "-1")
{
    if (lbl_dside.Content.ToString() != "-0.007827878")
    {
        far.visibility = Visibility.Visible;
    }
    else
    {
        far.visibility = Visibility.Hidden;
    }
}
else
{
    far.visibility = Visibility.Visible;
}

if (lbl_izrab.Content.ToString() == "1")
{
    if (lbl_dside.Content.ToString() != "-0.007827878")
    {
        fab.visibility = Visibility.Visible;
    }
    else
    {
        fab.visibility = Visibility.Hidden;
    }
}
else
{
    far.visibility = Visibility.Visible;
    fab.visibility = Visibility.Visible;
}

if (lbl_izside.Content.ToString() == "-1")
{
    if (lbl_dside.Content.ToString() != "-0.007827878")
    {
        fde.visibility = Visibility.Visible;
    }
    else
    {
        fde.visibility = Visibility.Hidden;
    }
}
else
{
    fde.visibility = Visibility.Visible;
}

if (cn.boton) { pt.visibility = Visibility.Visible; }
else { pt.visibility = Visibility.Hidden; }

// Se repite para cada botón hasta el 12

private void btn_conect_Click(object sender, RoutedEventArgs e)
{
    if (cn.Conectar())
    {
        btn_conect.Content = "Desconectar";
        CompositionTarget.Rendering += CompositionTarget_Rendering;
    }
}

private void btn_conect_Copy1_Click(object sender, RoutedEventArgs e)
{
    buscar_controles();
}
}
```

```
comboBox1.Items.Add(item.name);
}
}
else
{
    comboBox1.Text = "no hay controles conectados";
}
private void button1_Click_1(object sender, EventArgs e)
{
    buscar_controles();
}
}
```

Código de la aplicación WPF

```
namespace wpf_control
{
    public partial class MainWindow : Window
    {
        control_Ce cn;

        public MainWindow()
        {
            InitializeComponent();
            cn = new control_Ce();
            buscar_controles();
        }

        private void buscar_controles()
        {
            int cont = 0;
            comboBox1.Items.Clear();
            if (cn.get_dispositivos.Count > 0)
            {
                foreach (var item in cn.get_dispositivos)
            {
                cont++;
                if (cont == 1)
                {
                    comboBox1.Text = item.name;
                }
            }
            else
            {
                comboBox1.Text = "no hay controles conectados";
            }
        }

        private void CompositionTarget_Rendering(object sender, EventArgs e)
        {
            if (cn.get_palanca_derecha_ar_ab == 400)
            {
                buscar_controles();
                comboBox1.Items.Clear();
                comboBox1.Text = "no hay";
                return;
            }

            if (lbl_izrab.Content.ToString() != "-0.007827878" || lbl_dside.Content.ToString() != "-0.007827878")
            {
                fde.visibility = Visibility.Visible;
            }
            else
            {
                fde.visibility = Visibility.Hidden;
            }

            if (cn.boton) { pt.visibility = Visibility.Visible; }
            else { pt.visibility = Visibility.Hidden; }

            // Se repite para cada botón hasta el 12

            private void btn_conect_Click(object sender, RoutedEventArgs e)
            {
                if (cn.Conectar())
                {
                    btn_conect.Content = "Desconectar";
                    CompositionTarget.Rendering += CompositionTarget_Rendering;
                }
            }

            private void btn_conect_Copy1_Click(object sender, RoutedEventArgs e)
            {
                buscar_controles();
            }
        }
    }
}
```