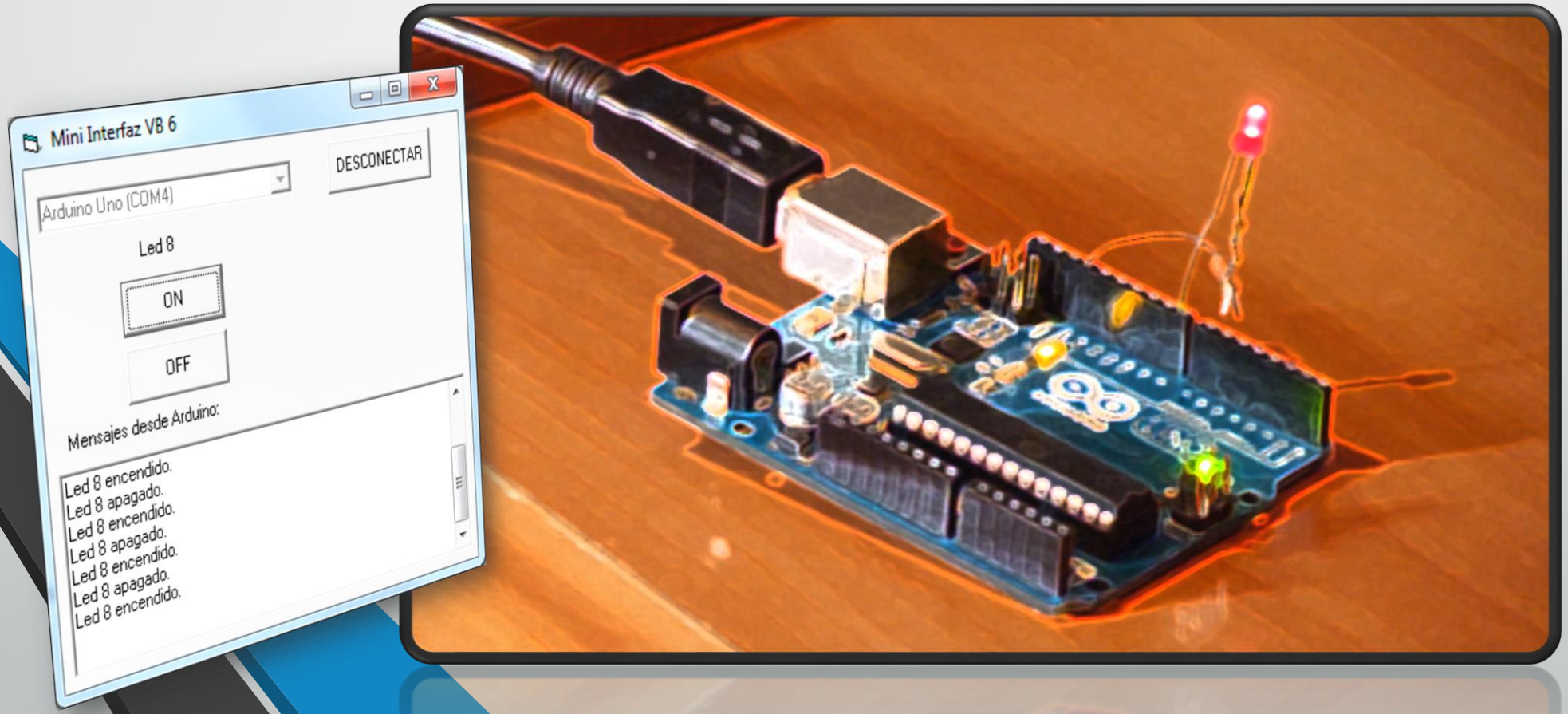
Tutorial

Interfaz Visual Basic 6 y Arduino



Índice

Presentación

- A pesar de hoy en día, se sigue usando Visual Basic 6 para los sistemas operativos modernos, aún en ciertos centros de enseñanzas de muchos países siguen con este lenguaje.
- Se presenta un tutorial donde podrás controlar Arduino mediante tu propia Interfaz programado con Visual Basic 6.
- Aún se usa mucho este lenguaje para la electrónica, manos a la obra, ánimos y adelante.

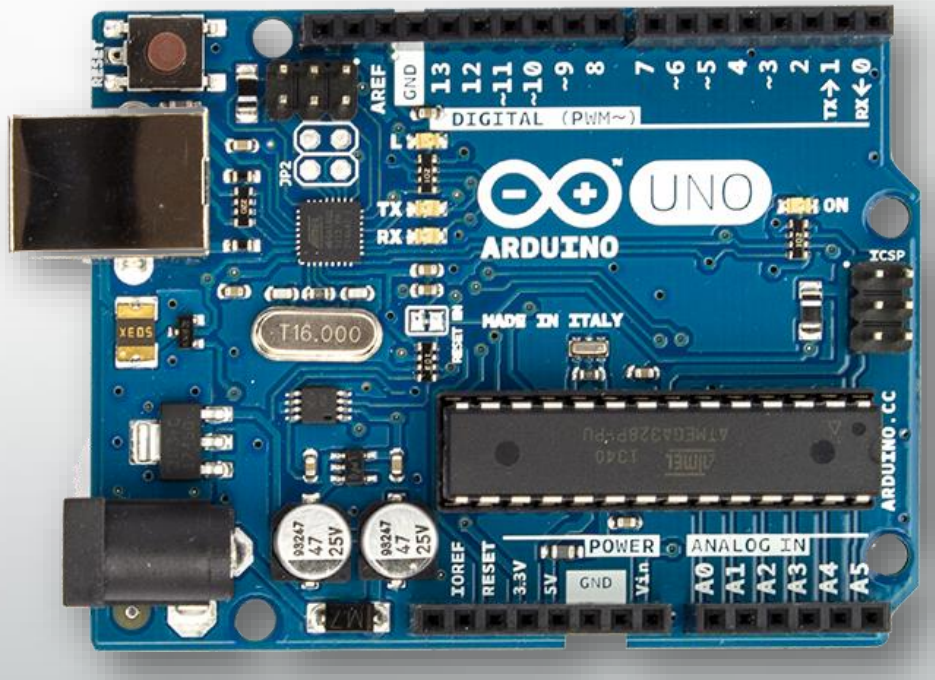
¿Qué necesitamos?

¿Qué necesitamos?

- Antes de trabajar en algo, hay que tener claro desde el principio que necesitamos realmente tanto en Hardware como en Software. Controlaremos con una Interfaz desde el PC con Arduino UNO r3.

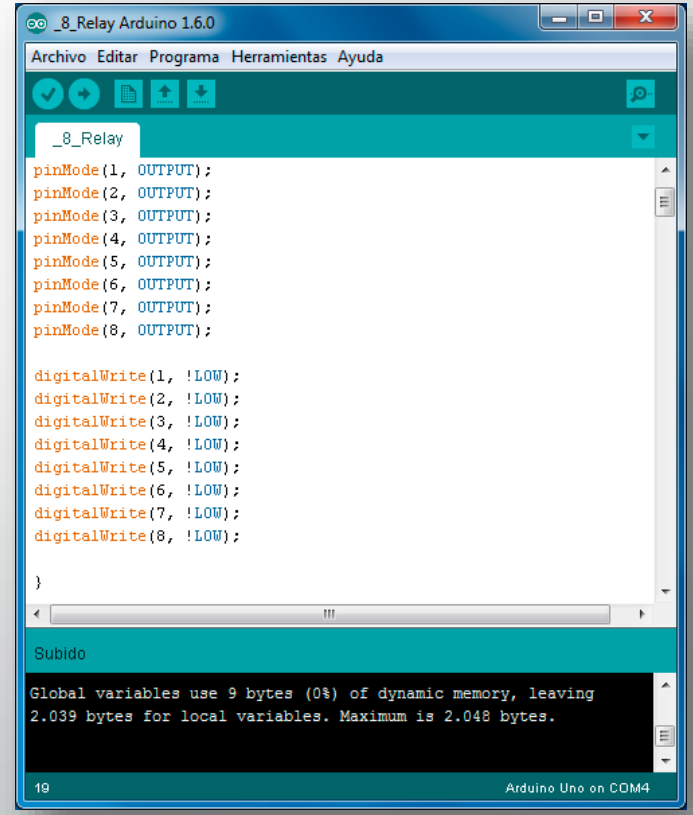
¿Qué necesitamos?

- Arduino UNO r3.
- Puedes usar Arduino el que dispongas.
- Escojo esta versión porque es el más usado y el más que tiene la mayoría gente a escala mundial para empezar.



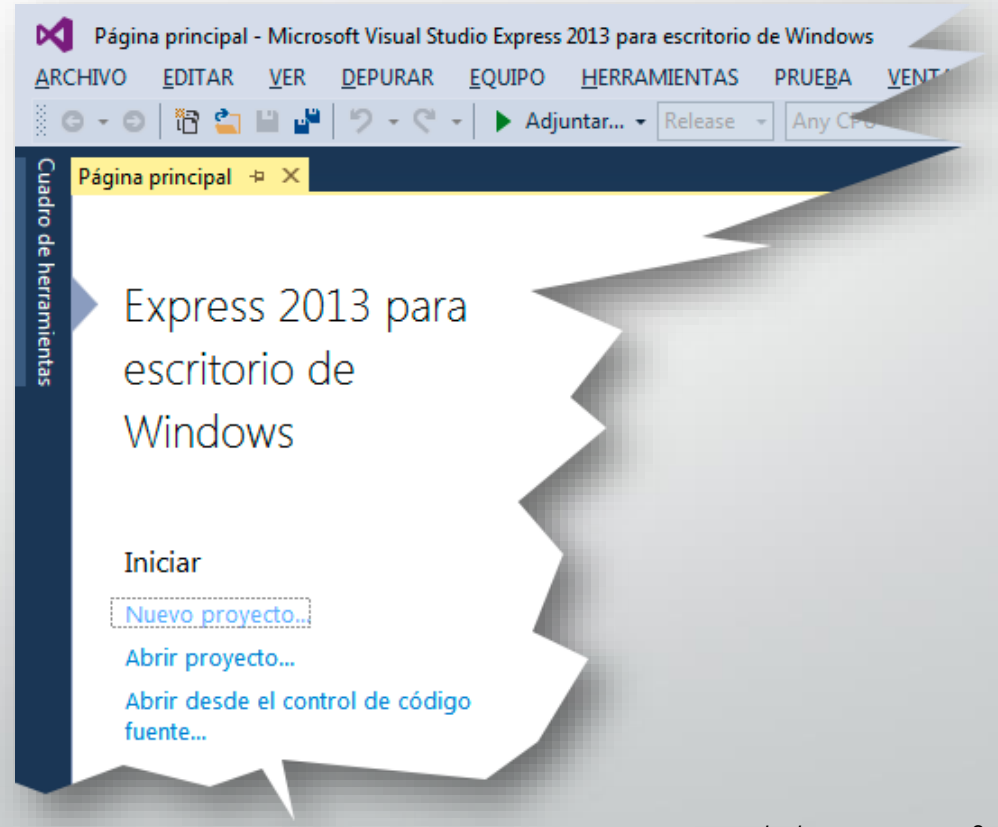
¿Qué necesitamos?

- Descargar Arduino IDE.
- http://arduino.cc/en/Main/Software



¿Qué necesitamos?

- Visual Studio Express 2013.
- Descargar **Express 2013 para escritorio de Windows.** *(En español).*
- http://www.visualstudio.com/es-es/products/visual-studio-express-vs.aspx
- También vale versiones 2005, 2010, 2012, 2013 y la Preview 2015 que veremos en Extras.



¿Qué necesitamos?

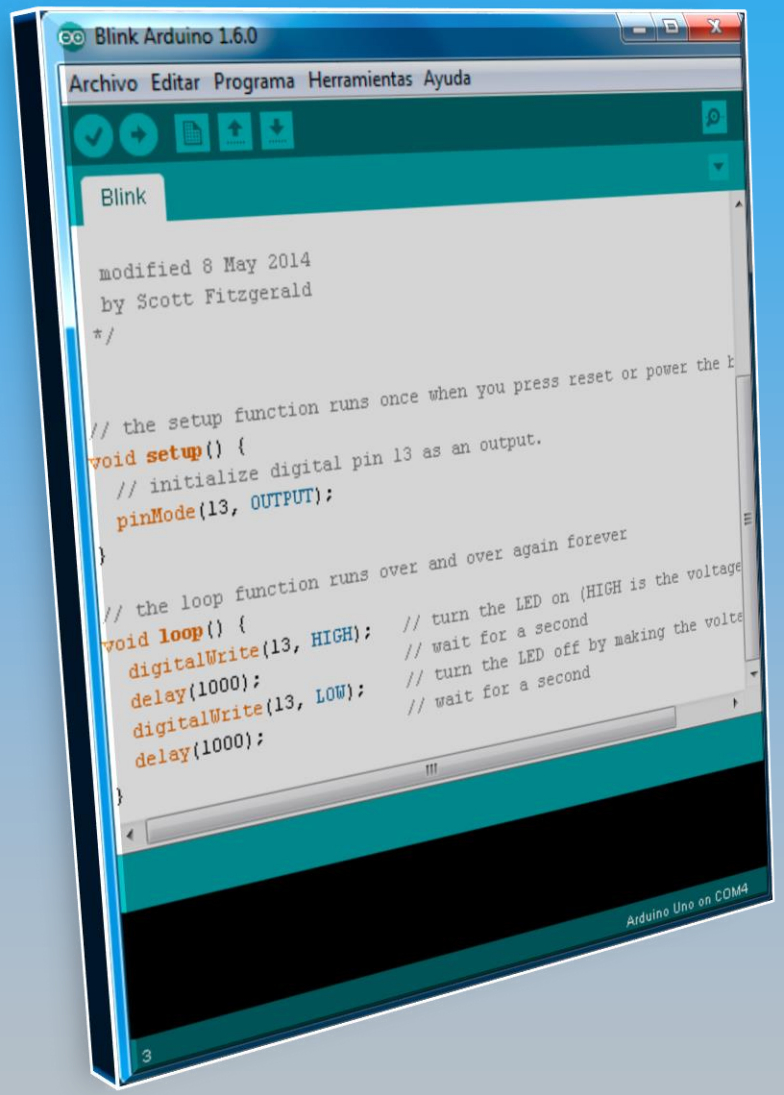
- Quizás le interese tener preparado este manual que si lo desea, puede mejorar tu interfaz un poco más detallado y completo como poner un reloj, hacer tu interfaz transparente y otras más opciones.
- http://electronica-pic.blogspot.com.es/2008/11/electrnica-pic.html
- A partir de la página 203 puedes leer los extras. Sólo hay ejemplos para Visual C#.
- Este tutorial fue el primero que creé sobre puertos series con Visual Studio .net en el 2008.

¿Qué necesitamos?

Descarga

- Puedes descargar de entrada todos los ejemplos y lenguajes.
- Se recomienda primero leer este tutorial antes de ver los ejemplos.
- Luego haz tu propio diseño de la interfaz.

- Fuente
- Fuente

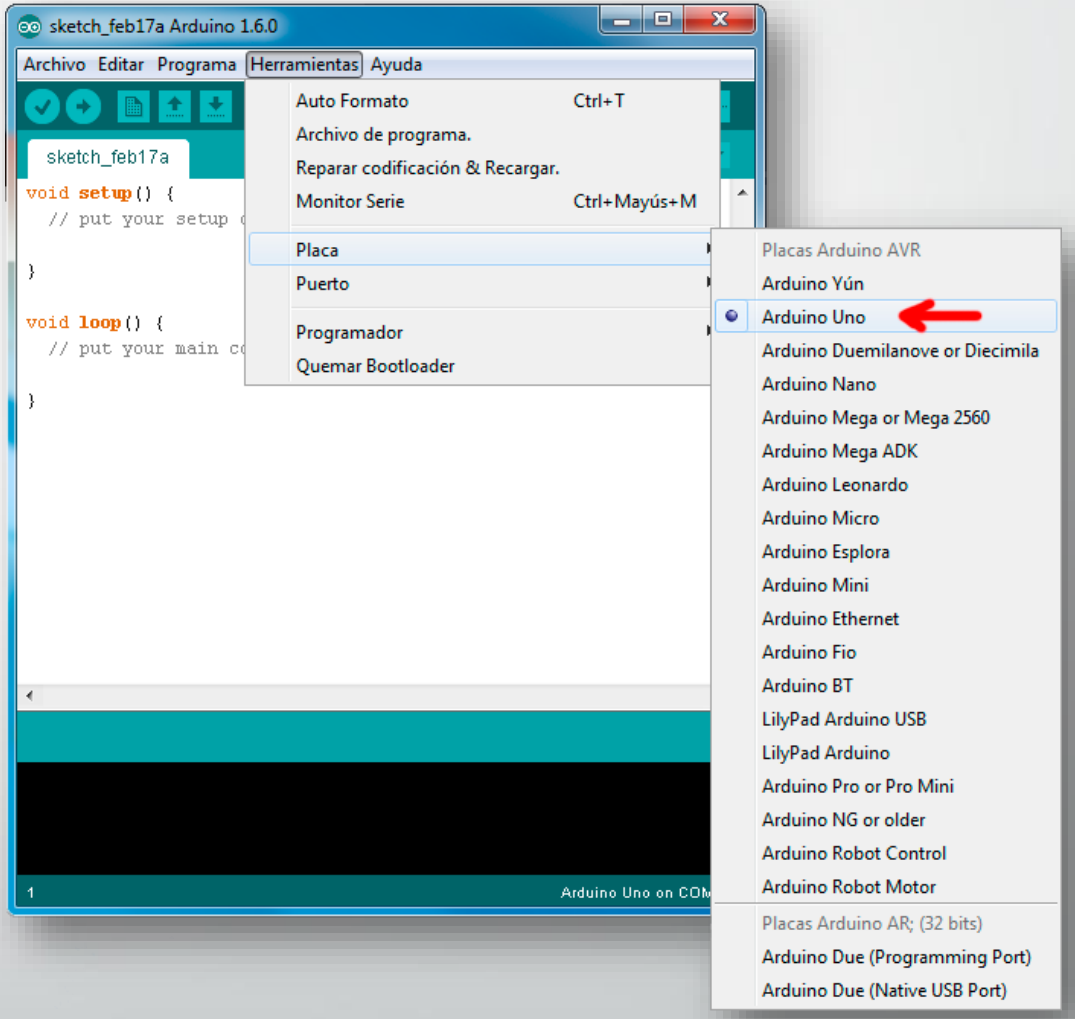


Configurar Arduino UNO

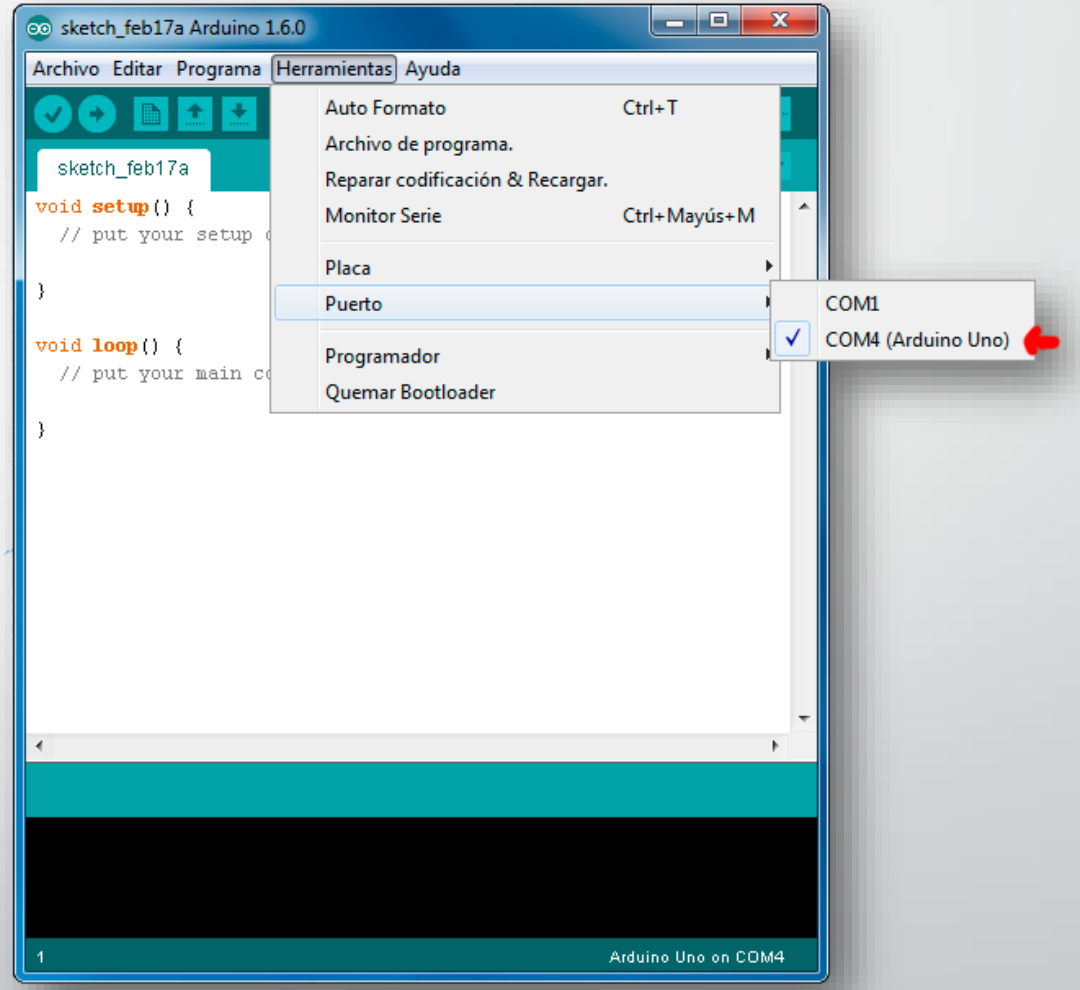
Configurar Arduino UNO

- Accedemos a la Web http://arduino.cc y hacemos clic en “Donwload”. En mi caso he descargado la versión “Windows Installer”.

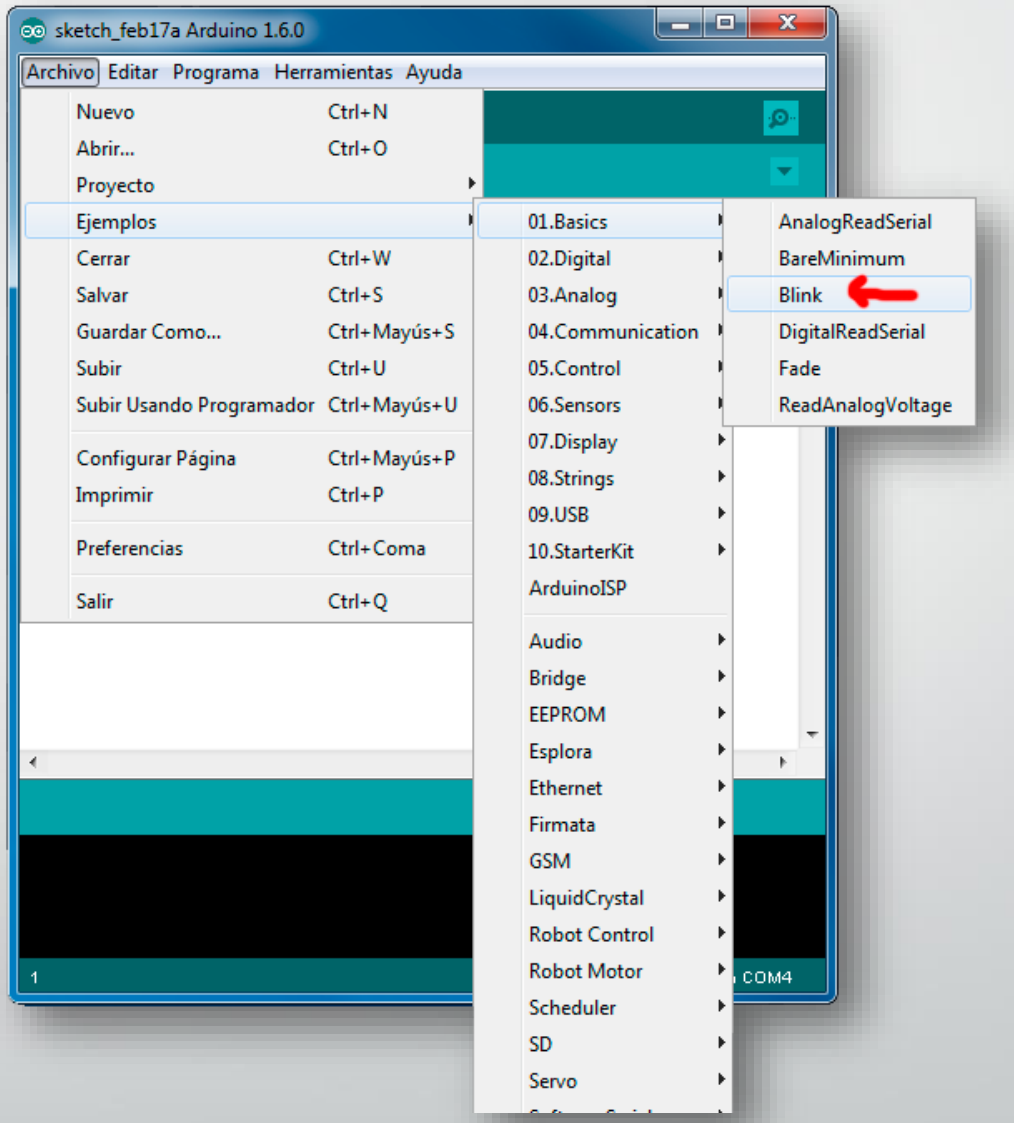
- Una vez ya descargado e instalado vamos a configurar la palca en Arduino UNO. Antes que nada, enchufa el cable USB al PC, la placa de Arduino UNO y luego ejecutas la Arduino IDE en este caso la versión es 1.6.0 desde que escribí este tutorial.
- Pulsas "Herramientas → Placa → Arduino UNO".



- Nos aseguramos que tengas el puerto correspondiente de Arduino.
- En este caso es el puerto COM4.



- Para saber que todo está en orden abriremos uno de los ejemplos y famoso parpadeo del Led 13 que te viene incluido en la placa Arduino UNO.
- Pulsa "Archivo →Ejemplos → 01.Basics → Blink".



Configurar Arduino UNO

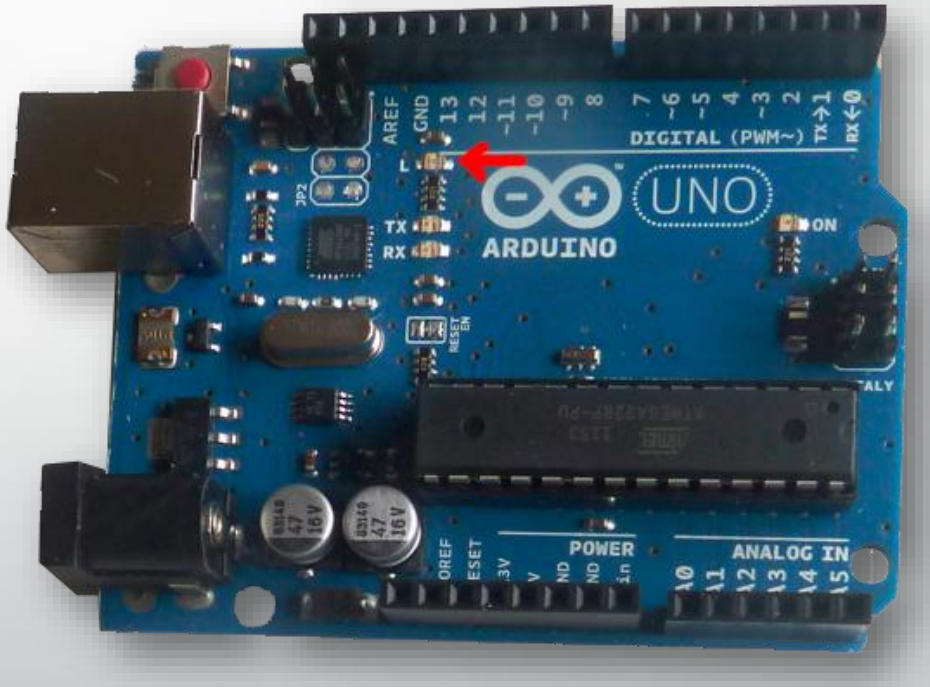
```
void setup() {

pinMode(13, OUTPUT); // Inicializa pin 13 como salida digital.

}

// Aquí se repite el cliclo una y otra vez.

void loop() {

 digitalWrite(13, HIGH);  // Led encendido.

delay(1000);  // Retardo de un Segundo.

digitalWrite(13, LOW); // Led se apaga;

 delay(1000);  // Espera un Segundo.

}
```

Configurar Arduino UNO

- Está más que verificado este código. Pulsa "Subir" para compilar el código y guardarlo en Arduino.
- Si no has pulsado "Verificar" antes de compilar verifica código, compila y lo guarda en el AVR de Arduino.
- Abajo de Arduino IDE 1.6.0 aparecerá cualquier información como cantidad de datos usados en portentajes, incluido errores entre otras cosas.

Configurar Arduino UNO

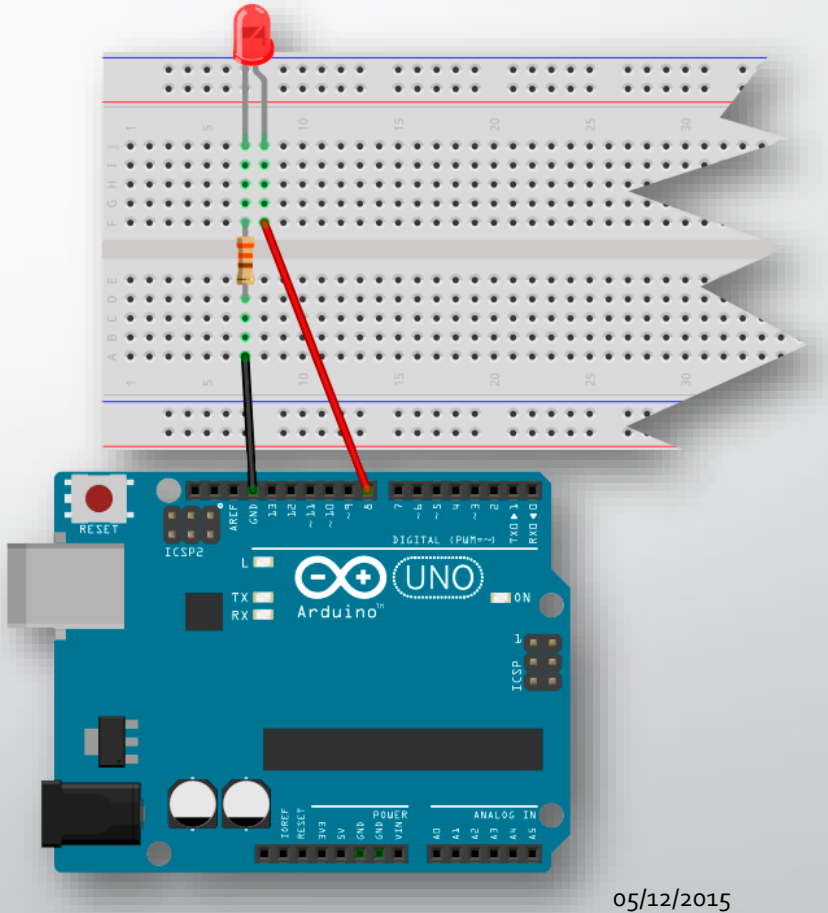
- Fíjate que el Led 13 donde indica la flecha roja está parpadeando.



Esquema boceto

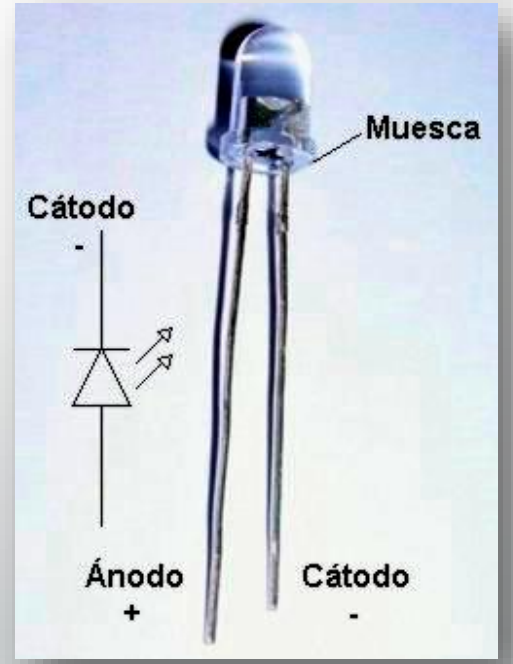
Esquema boceto

- Tal como muestra en la imagen, usaremos una resistencia de 330 Ohmios y un diodo Led rojo conectado a Arduino UNO r3 de los pines GND y el pin 8.
- Si tienes dudas, antes de montar el circuito, pasa a la página siguiente.



Esquema boceto

- Muestra el diodo Led y su simbología.
- El pin más corto es el Cátodo, también se escribe con K, Kátodo, es el negativo del Led.
- Si se encuentran un Led con los pines cortados de igual medida, hay que fijarse en la muesca que representa el Cátodo del Led.



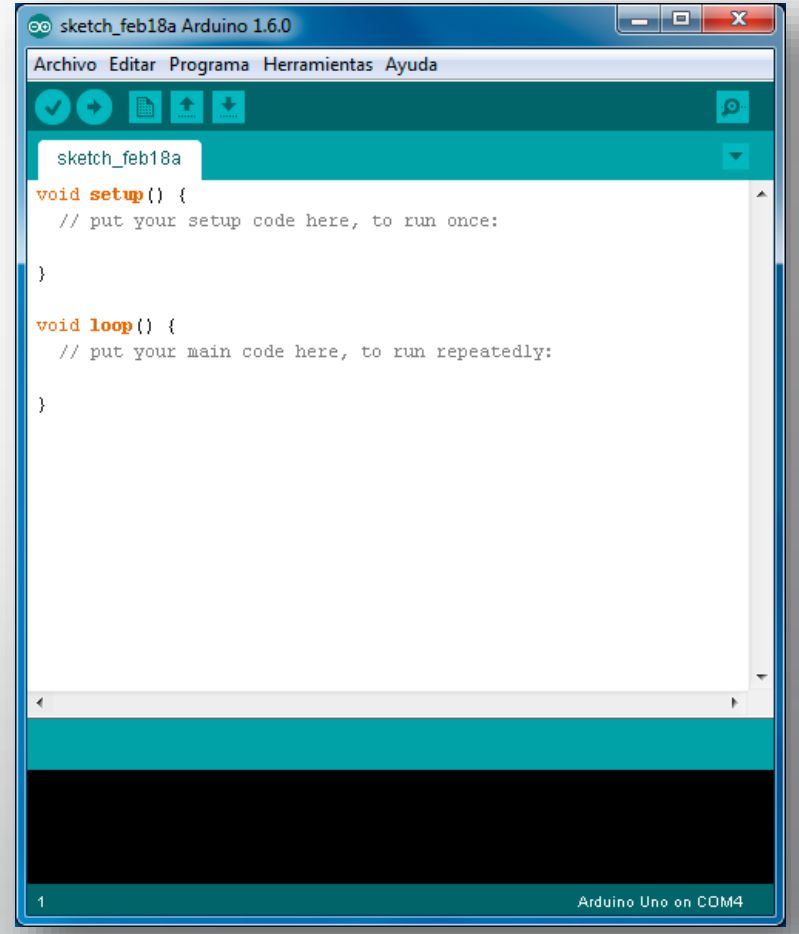
Programación Arduino IDE

Programación Arduino IDE

- Escribimos el código.
- Este pequeño código permite leer por el puerto USB que en realidad es puerto serie emulado para apagar y encender un Led.

```
char caracter;
String comando;

void setup(){
	pinMode(8, OUTPUT); // Configuramos el pin 8 como salida.
	Serial.begin(115200); // Iniciamos el puerto serie a 115200 baudios.

}

void loop(){ // Leemos carácter a carácter lo que recibimos por el puerto serie y concatenamos uno a uno a formar una cadena.

	while (Serial.available()>0){
	caracter= Serial.read();
	comando.concat(caracter);
	delay(10);
}

/* Cuando tengamos la cadena acabada, comprobamos el valor aquí abajo. Si no se encuentra, lo ignorará. Ya podemos encender un Led, motores, ventiladores, bombillas y otros dispositivos que tengamos conectado mediante relés o otros medios. */

if (comando.equals("Led_ON") == true){ // Si en la cadena de caracteres se incluye la palabra "Led_ON".
	digitalWrite(8, HIGH); // Enciende el Led.
	Serial.println("Led encendido.");
}

if (comando.equals("Led_OFF")== true){ // Si en la cadena de caracteres se incluye la palabra "Led_OFF".
	digitalWrite(8, Low); // Apaga el Led.
	Serial.println("Led apagado.");
}

comando=""; // Limpiamos la cadena para volver a recibir el siguiente comando.
}
```

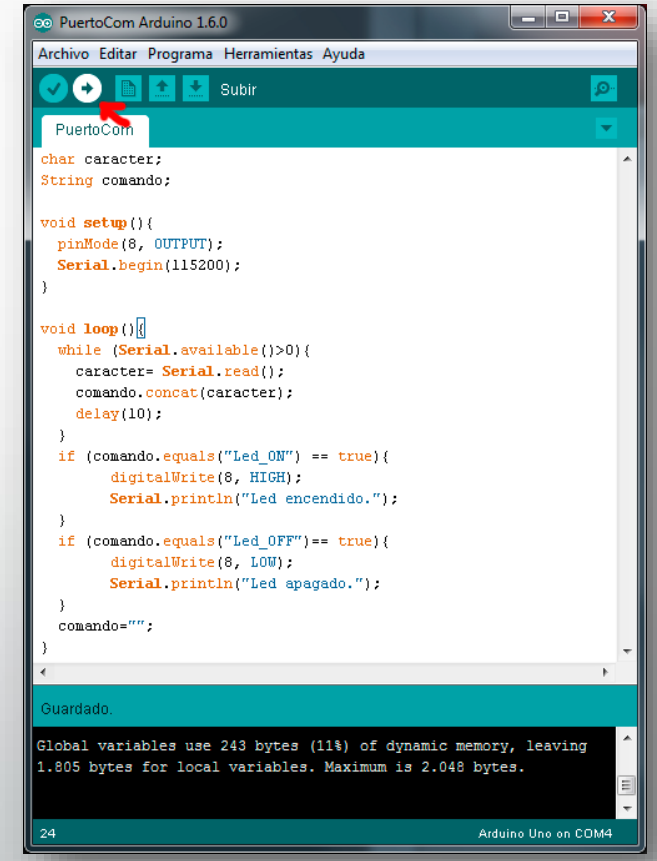
Programación Arduino IDE



- Al crear un archivo nuevo viene así predeterminado.

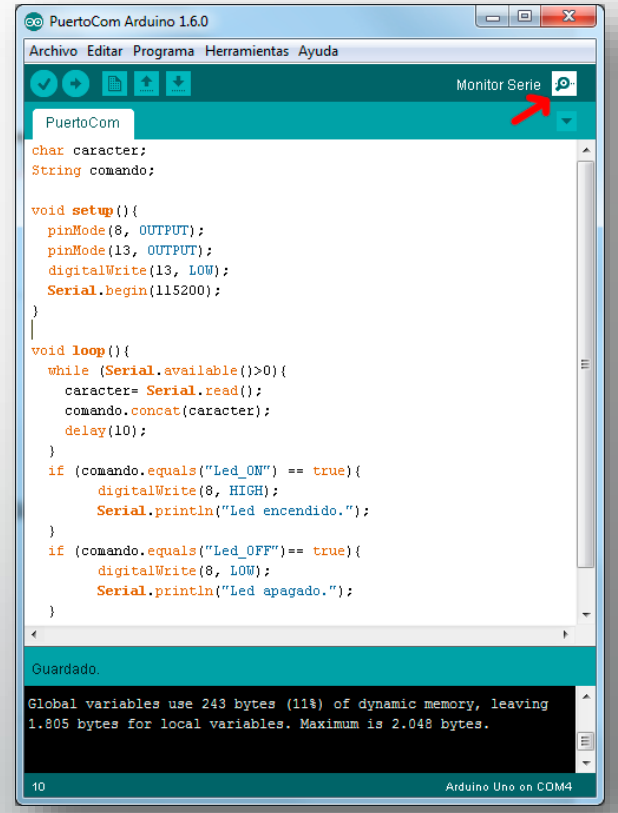
Programación Arduino IDE

- Escribimos el código como muestra en la imagen.
- Teniendo Arduino UNO conectado con el cable USB pulsamos el botón “Subir” y grabamos nuestro programa.



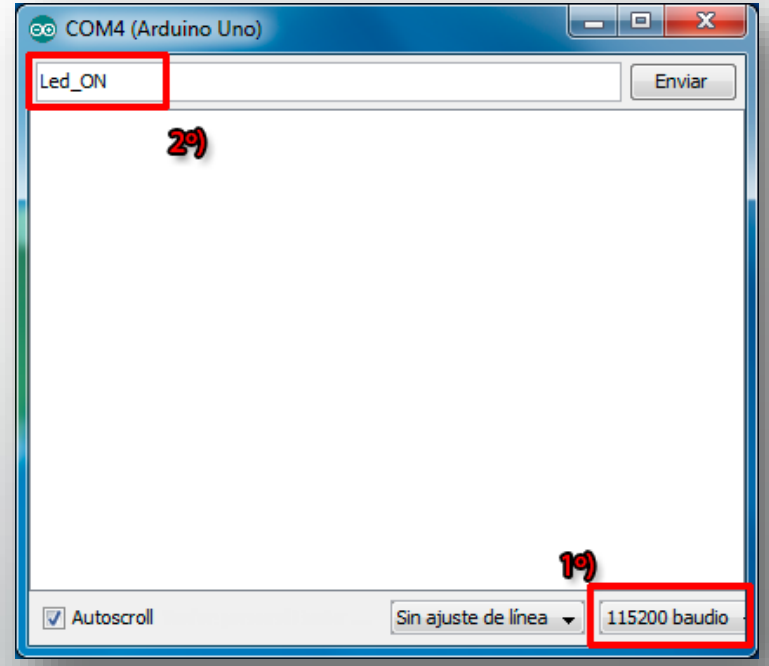
Programación Arduino IDE

- Pulsamos el botón "Monitor Serie" para enviar comandos.



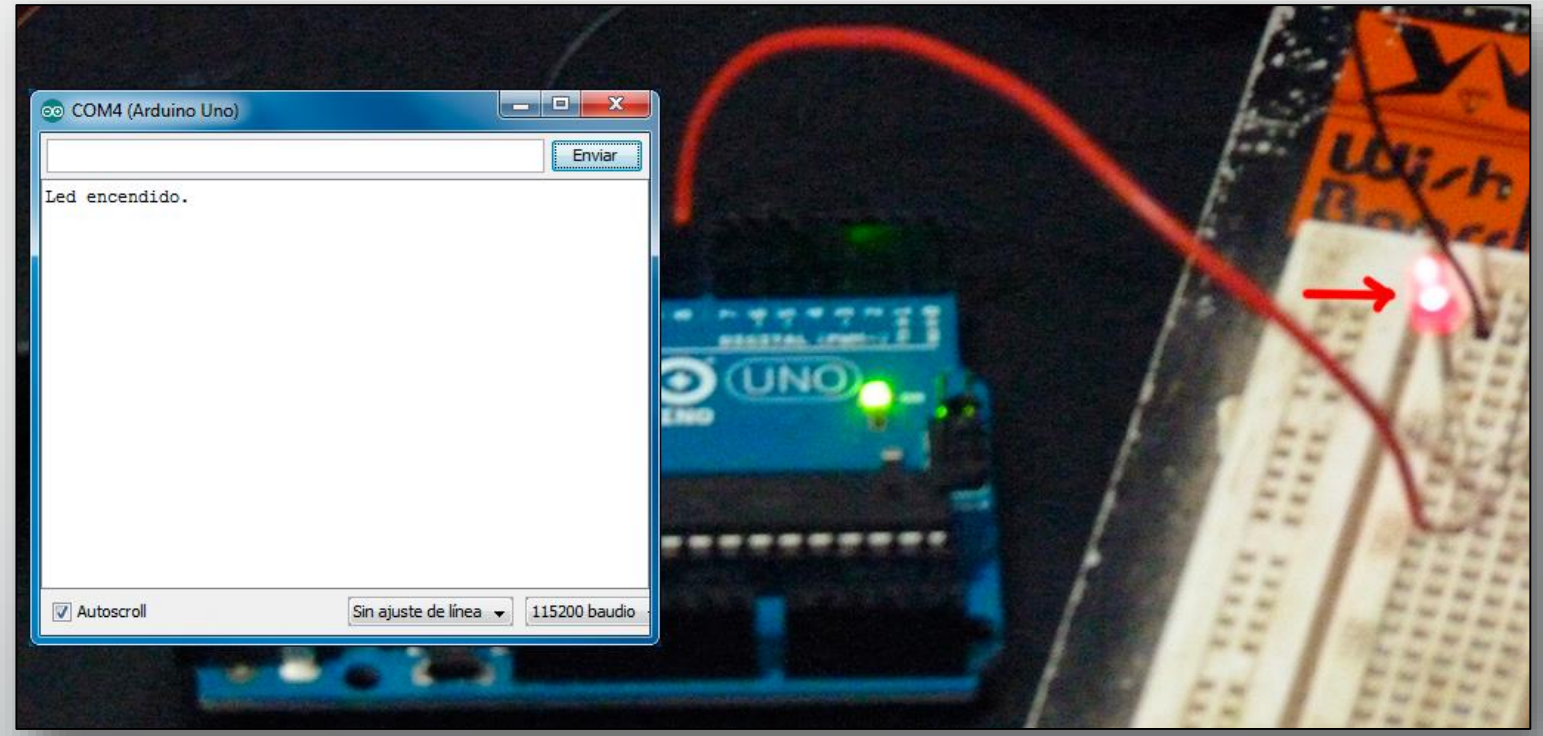
Programación Arduino IDE

- 1º) poner los baudios a 115200 como tenemos en el programa en Arduino IDE.
- 2º) Introducir los comandos **Led_ON** para encender el Led y **Led_OFF** para apagar, luego pulsas "Enviar".



Programación Arduino IDE

- Muestra el mensaje: "Led encendido.".
- Luego pruebas con el comando Led_OFF.
- Si funciona, estamos preparado para hacer la interfaz bajo Windows, Linux u otro Sistema Operativo.

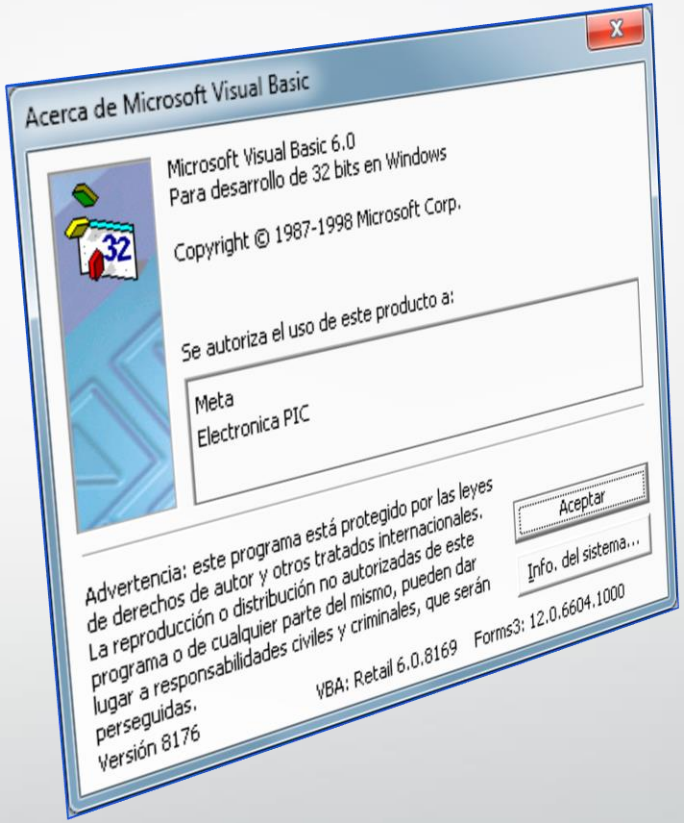


Visual Basic 6

Interfaz puerto serie

Visual Basic 6

En este caso estoy trabajando bajo Windows 7 de 64 bits y VB 6.



Objetivo:

Diseñar una pequeña interfaz desde cero.

1) Insertar 3 botones para dos Led, uno del pin 8 llamado ON, otro OF y el otro CONECTAR.
2) Insertar label o etiquetas y un comboBox.
3) Insertar un componente Text en el formulario para recibir mensajes desde Arduino.
4) Insertar y configurar el componente o objeto MSComm1.
5) Enviar comandos con los botones desde la Interfaz que creamos hacia Arduino.
6) Recibir mensajes desde Arduino.

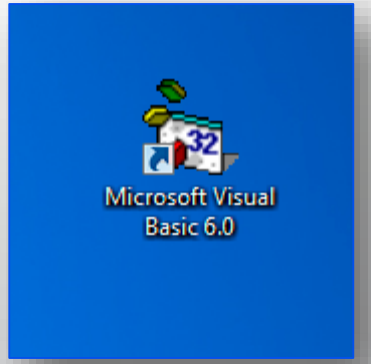
Visual Basic 6

- A día de hoy, hay muchas personas que usan Visual Basic 6. Por el 2008, a pesar de advertir que no haré tutoriales sobre VB 6 ya que es muy obsoleto y me centré en el VB .net, se sigue usando bastante en centros de enseñanzas de muchos países.
- Da igual que recomiende el VB .net, hay mentes empeñadas en VB 6.
- También usan equipos muy viejos con puertos series bajo Windows 95/98 para muchos proyectos.
- También funciona muy bien para el Windows XP aunque ya existe el VB .net de la época.

- Hoy en día se usa mucho VB 6 para pequeños proyectos relacionado con la electrónica.
- En este caso uso Windows 7 de 64 bits y funciona muy bien hasta con puertos virtuales para controlar Arduino.
- Entonces, hagamos una interfaz sencillo con Visual Basic 6 que tanto me han pedido durante años.

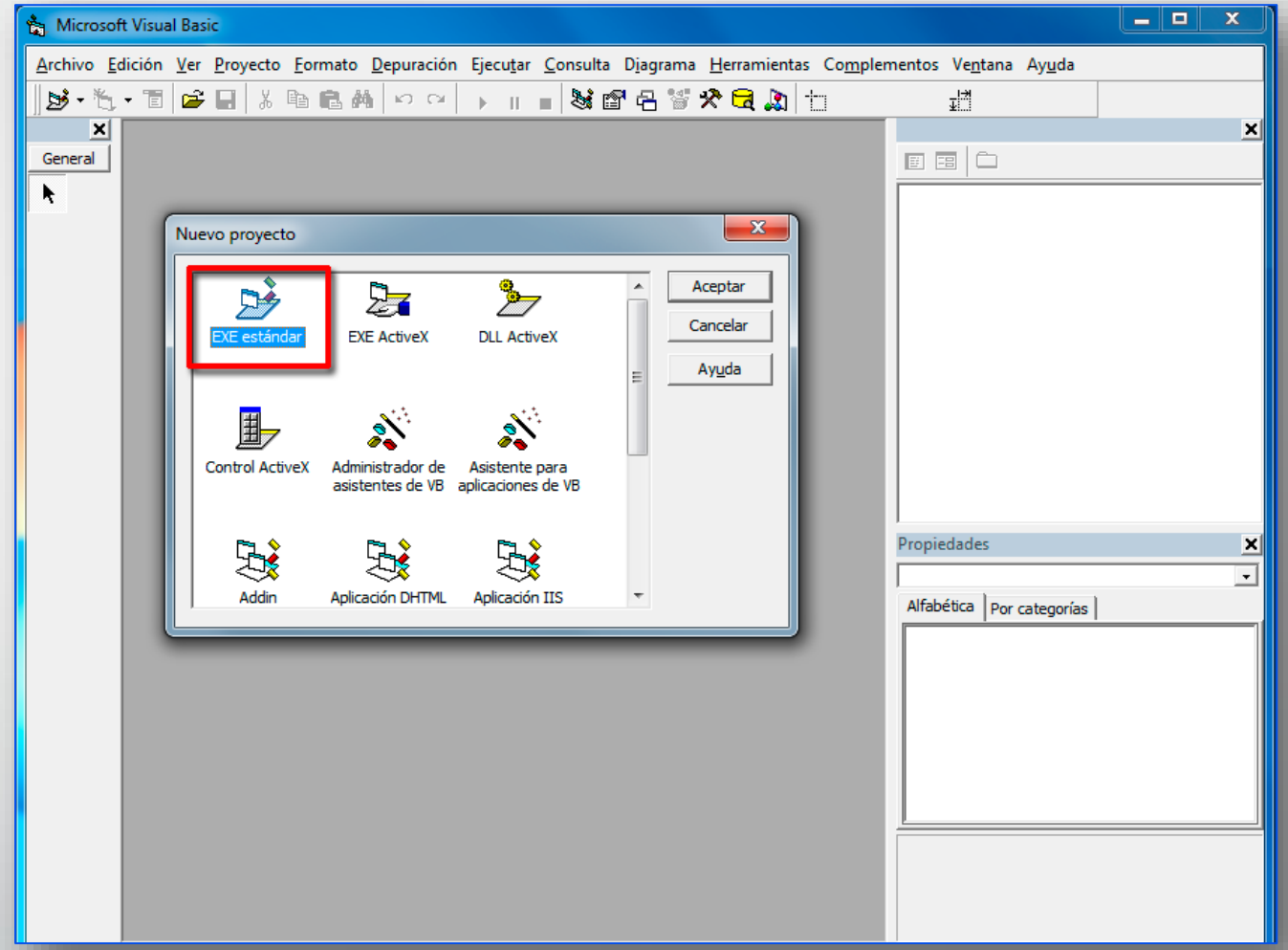
Visual Basic 6

- Ejecutaremos Visual Basic 6 en modo Administrador.



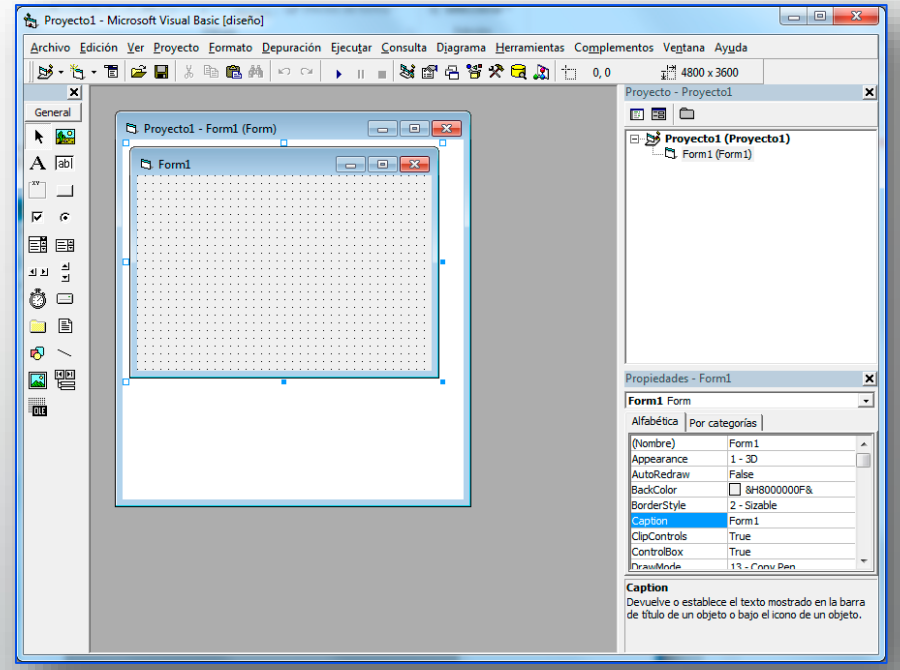
Visual Basic 6

- Seleccionamos “EXE estándar”.
- Finalmente pulsamos “Aceptar”.



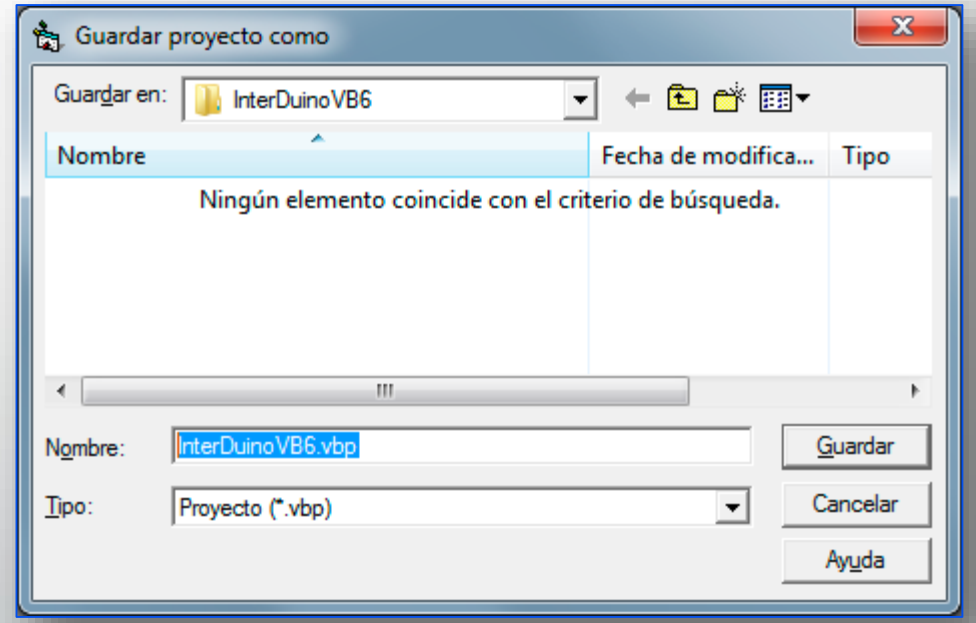
Visual Basic 6

- Muestra el formulario donde configuramos las propiedades e incluimos botones más seguimos con los códigos de programación.



Visual Basic 6

- Guardamos el proyecto en "Archivo" → "Guardar proyecto".
- Se abre una ventana, antes creamos una carpeta o directorio llamado "InterDuinoVB6".
- En el directorio llamamos el formulario como nombre **InterDuinoVB6.frm** y pulsamos "Guardar".
- Sale otra ventana, le he puesto el nombre del proyecto **InterDuinoVB6.vpb** y pulsamos "Guardar".



Visual Basic 6

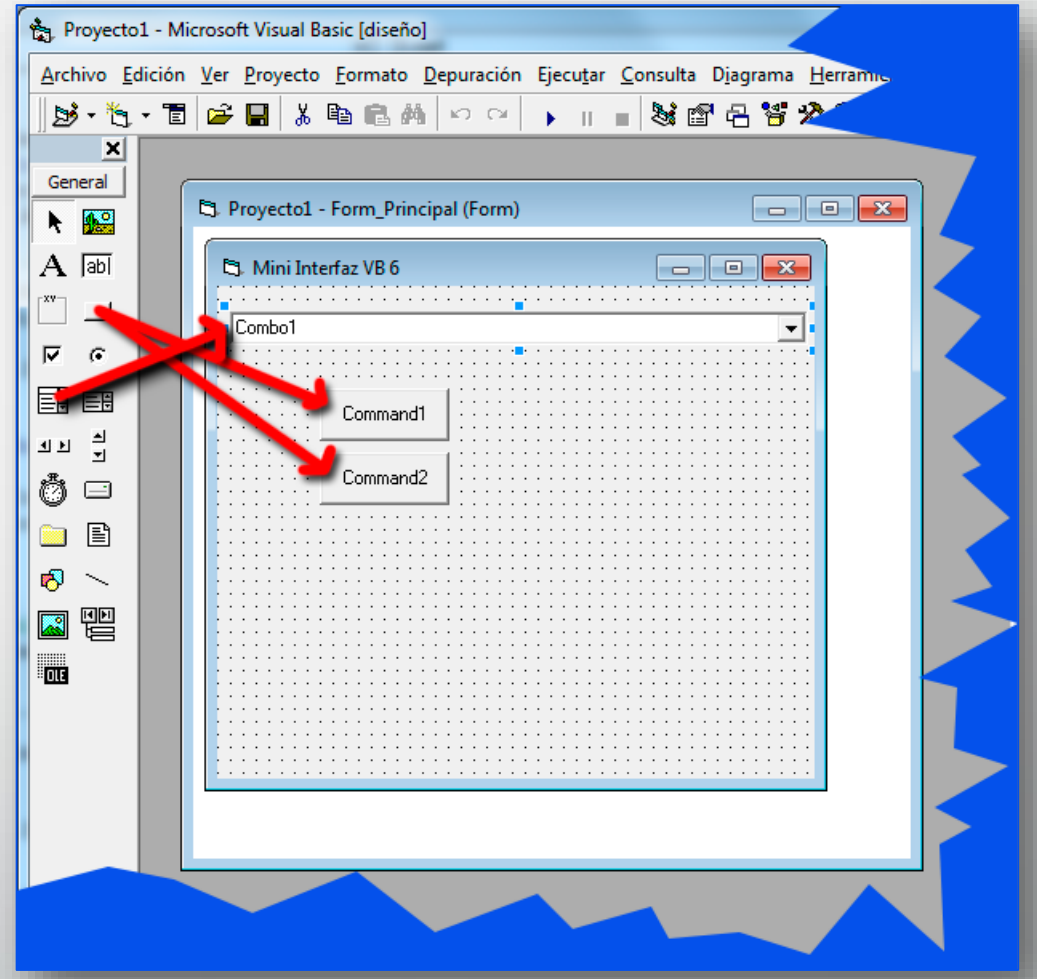
Propiedades

- Fijándonos en la página anterior. Cambiamos las propiedades del formulario tal como muestra aquí.
- **Caption,** es el texto donde ponemos el título de la pantalla.
- **StartUpPosition** donde quieres que se muestre la ventana de nuestro interfaz. En este caso lo dejamos en el centro.

Propiedades	Cambia a
(Nombre)	Form_Principal
Caption	Mini Interfaz VB 6
StartUpPosition	2 - CenterScreen

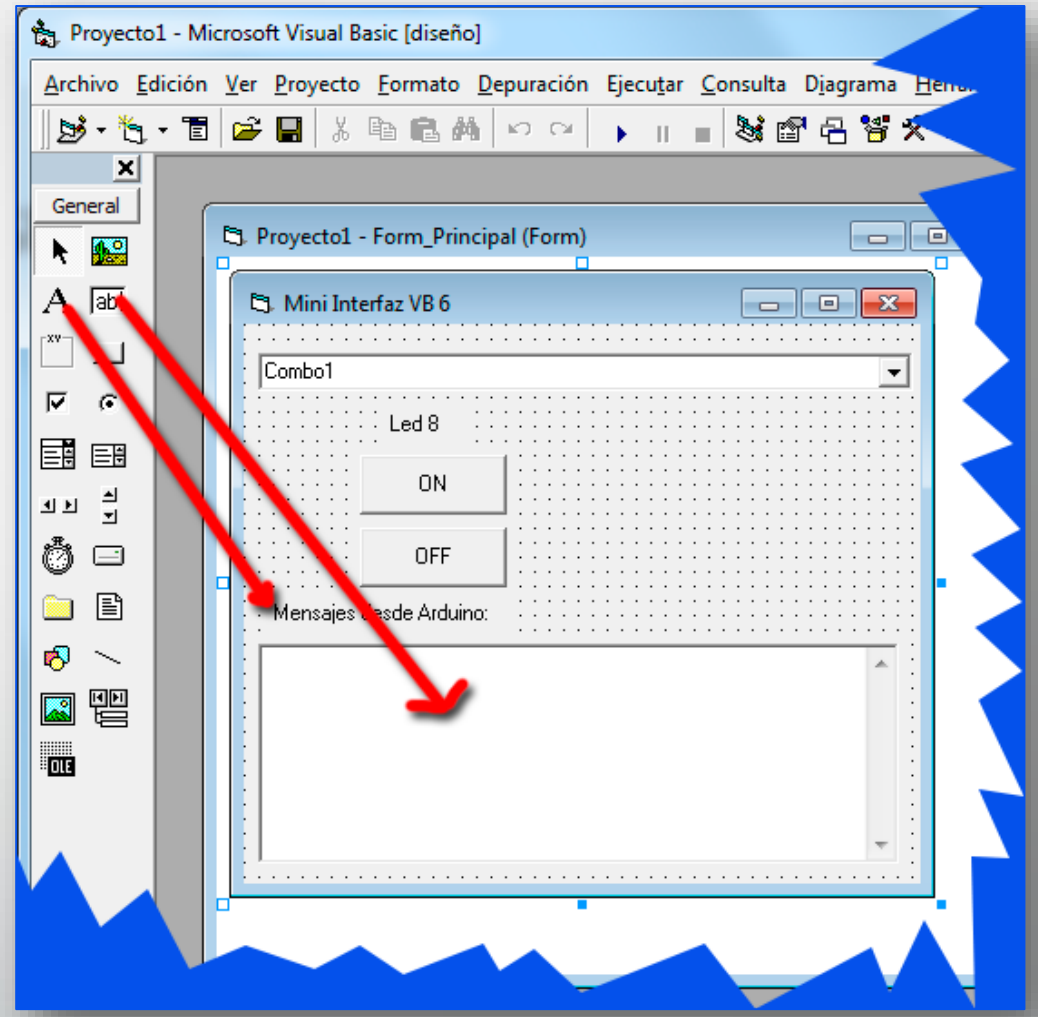
Visual Basic 6

- En el "Cuadro de herramientas", seleccionamos dos "Command" que son botones y un "ComboBox".
- El nombre interno del comboBox lo he llamado **ComboBoxCOM,** así que si sigues el ejemplo, no lo olvides, está en la propiedad (Name).



Visual Basic 6

- Colocamos más botones y lo colocamos más o menos como muestra la imagen.
- Luego arrastramos el objeto Text en el cuadro, aquí nos llegará los mensajes de textos como en el "Monitor Serie" desde Arduino.
- En sus propiedades del Text le ponemos Multiline en True ya que será de muchas líneas la entrada de datos.



Visual Basic 6

Propiedades

- En el cuadro de texto Text para recibir mensajes de textos desde Arduino.
- **MultiLine** para múltiples líneas de texto, uno debajo del otro.
- **ScrollBars** para crear una barra de desplazamiento vertical.
- **Loked** para no modificar los textos de entrada por el puerto serie, se queda en modo lectura.

Propiedades	Cambie a
(Name)	Text_Mensajes
MultiLine	True
Text	
ScrollBars	2 - Vertical
Loked	True

Visual Basic 6

Propiedades

- En cada botón configuramos las propiedades.
- No olvidar que seleccionamos el primer botón, así con cada uno de ellos.

Propiedad	Cambie a
Caption	ON
(Name)	Command_Led_8_ON
Enabled	False

Visual Basic 6

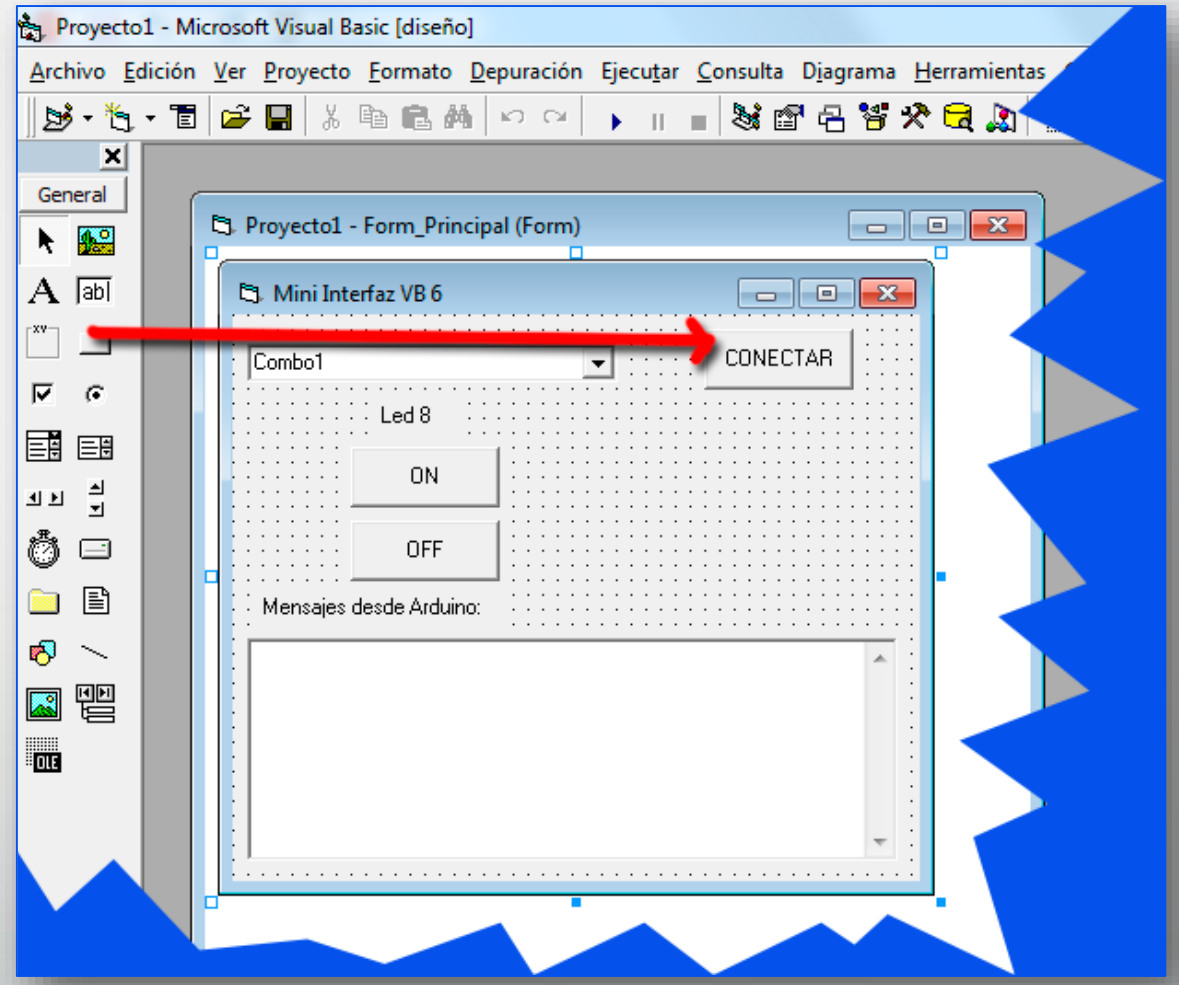
Propiedades

- El otro botón lo mismo pero en OFF.

Propiedad	Cambie a
Caption	OFF
(Name)	Command_Led_8_OFF
Enabled	False

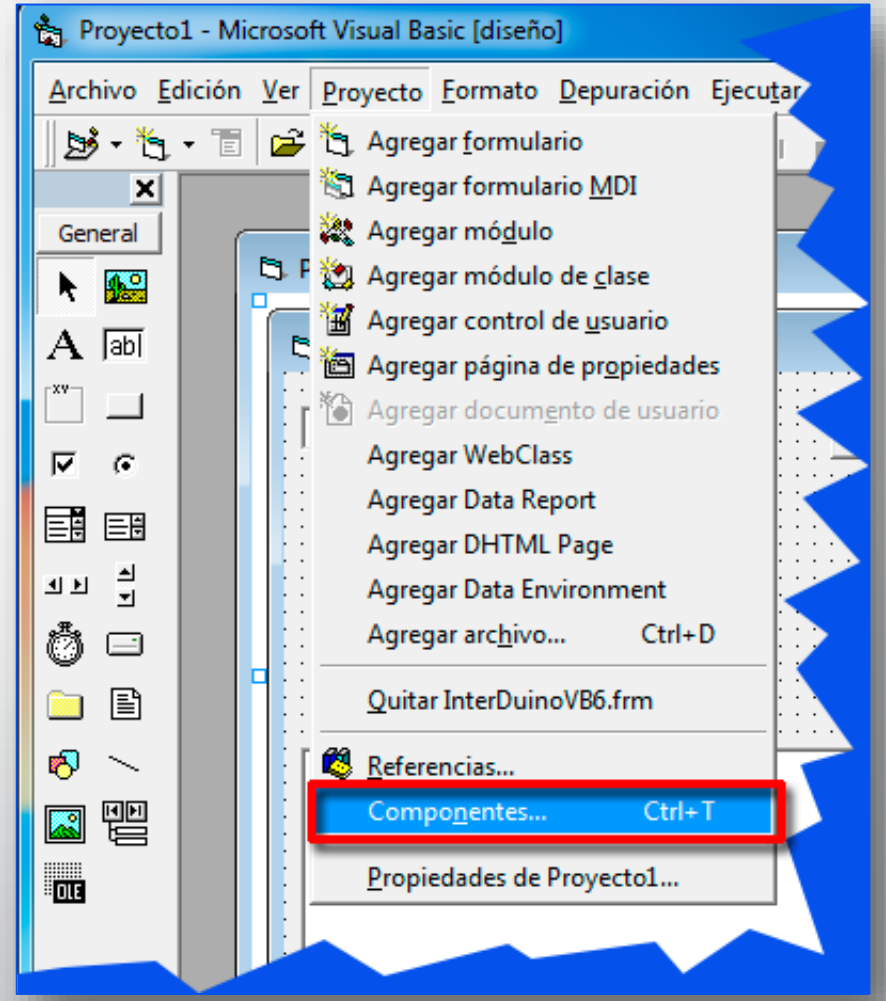
Visual Basic 6

- Redimensionamos el ComoBox a la propiedad Width (ancho) a 3015.
- Insertamos un botón y lo llamaremos **Command_CONECTAR** de la propiedad (Name).
- En "Caption" del botón ponemos **CONECTAR**.



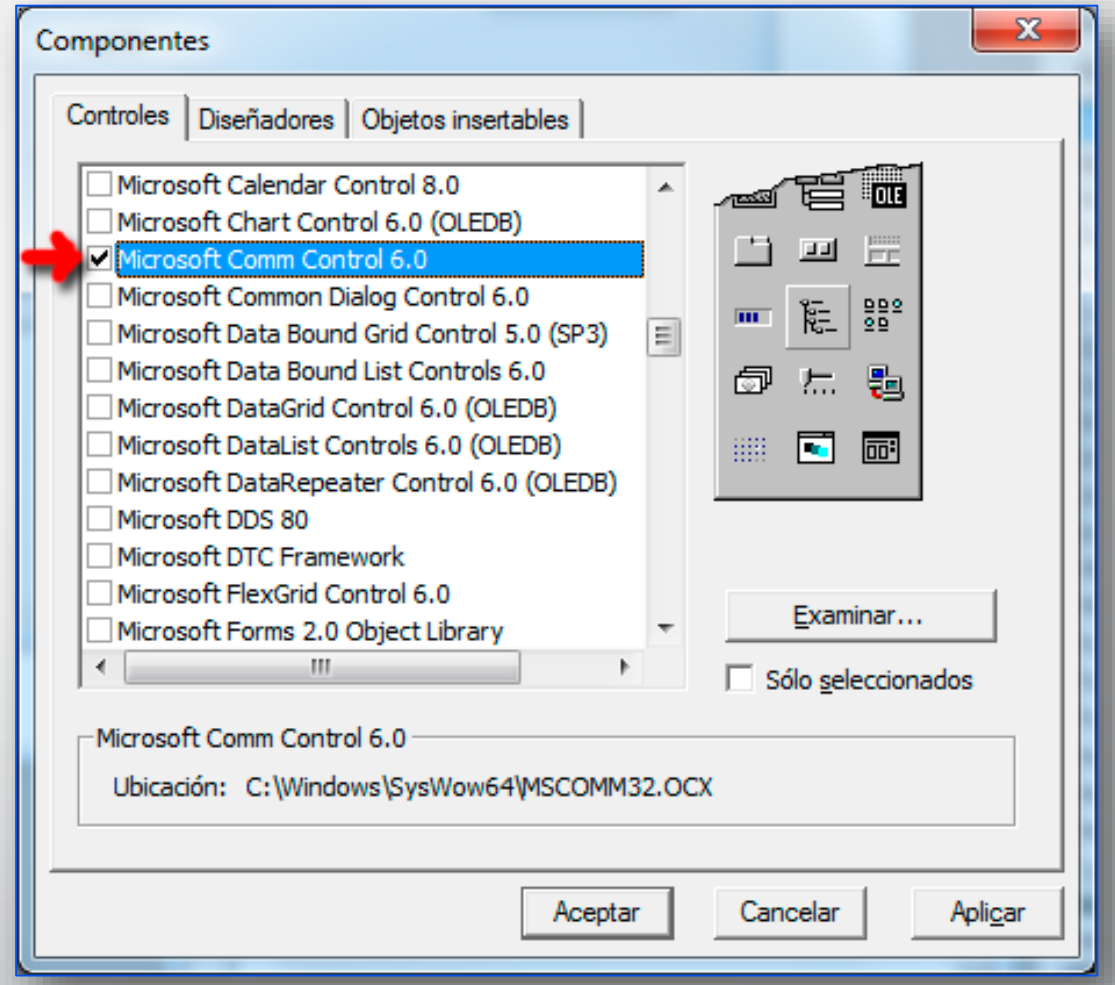
Visual Basic 6

- Añadimos dos componentes más.
- En la barra de arriba pulsamos "Proyecto" → "Componentes".



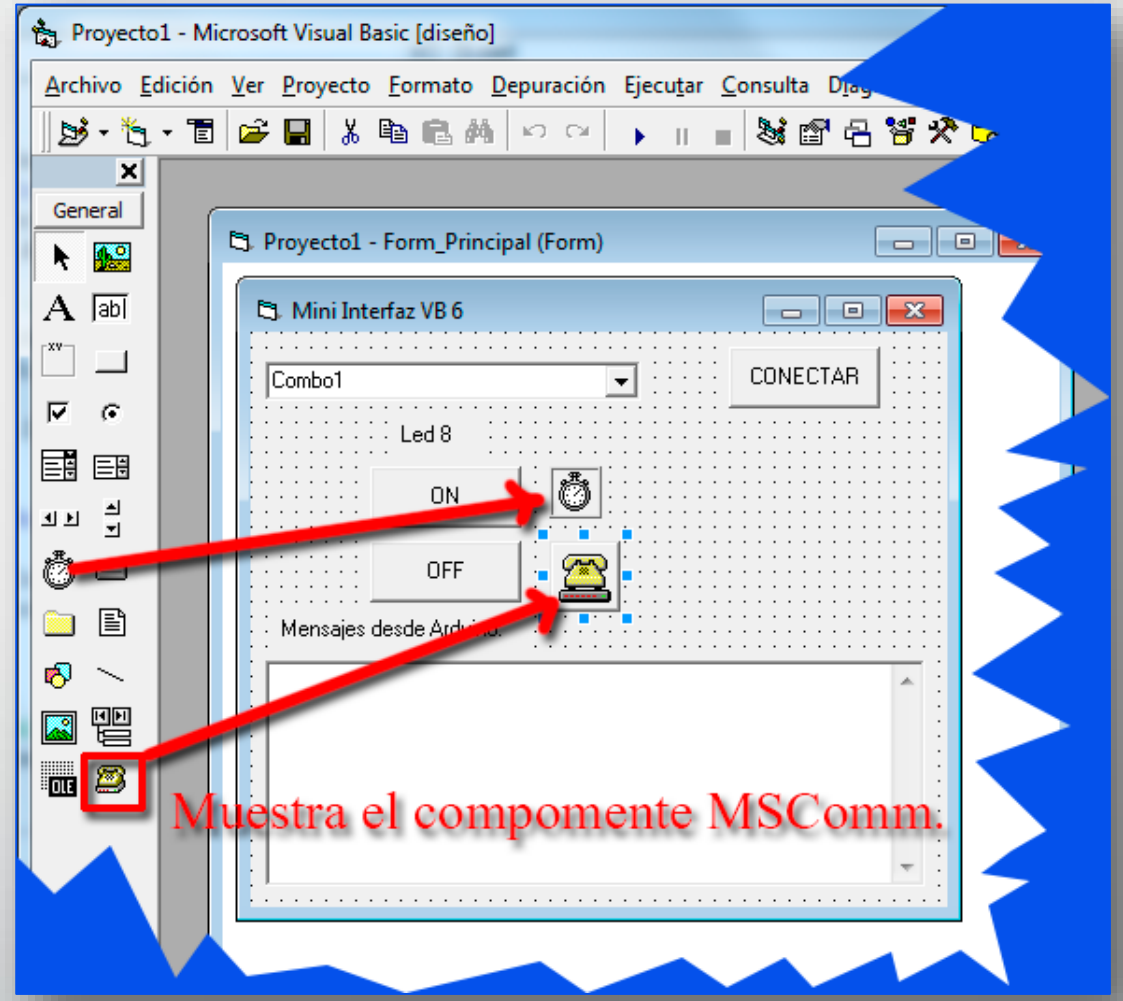
Visual Basic 6

- Se abre la ventana "Componentes".
- En la pestaña "Controles" seleccionamos el componente "Microsoft Comm Control 6,o" que es para manejar el puerto serie.
- Pulsamos "Aplicar" y luego "Aceptar". Como hemos elegido solo un componente, se puede pulsar solo "Aceptar"directamente.



Visual Basic 6

- Verás un componente nuevo en forma de teléfono clásico amarillo para controlar el puerto serie.
- Añadimos un timer y el MSComm en el formulario.
- Al ejecutar la aplicación o programa no ve mostrará estos componentes en forma visible.



Visual Basic 6

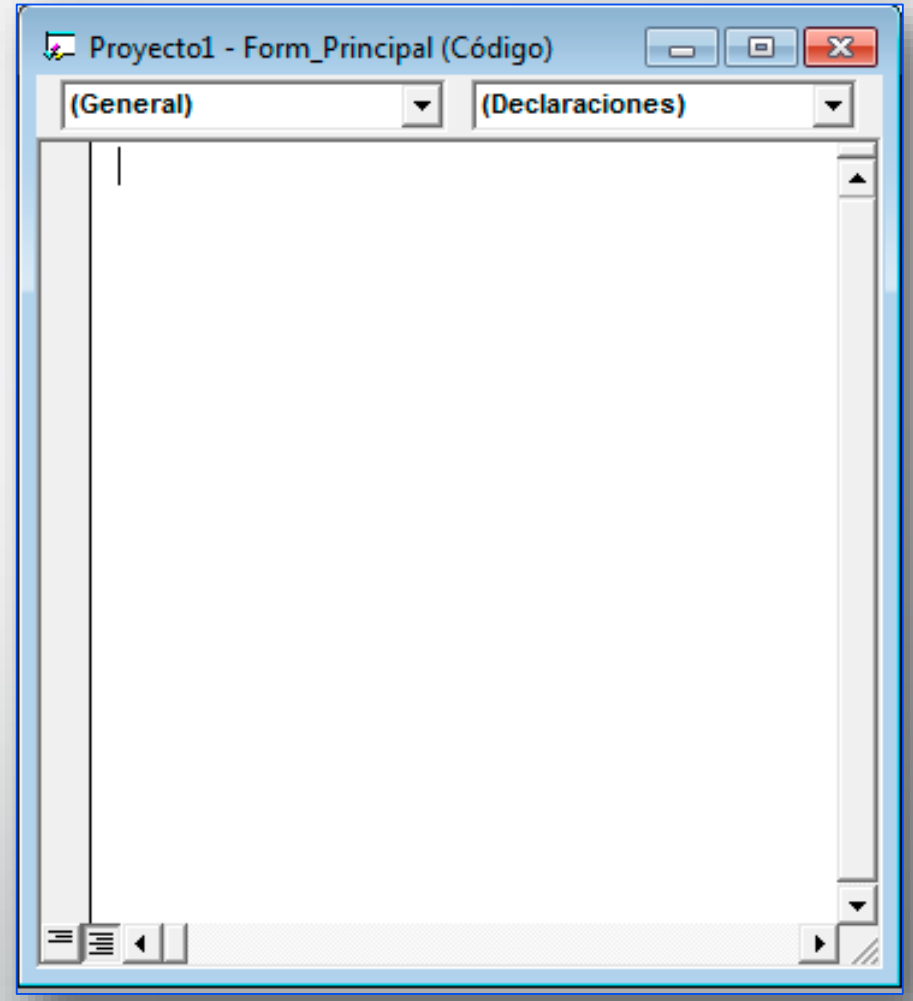
Propiedades

- Seleccionamos la propiedad MSComm y cambiamos sus propiedades indicado en el cuadro de al lado.
- No olvidar que los valores del Settings tiene que ser el mismo en el Arduino para que haya comunicación.
- **RThreshold** Devuelve o restablece el número de caracteres al recibir.

Propiedad	Cambie a
(Name)	MSComm_Puerto_Serie
CommPort	1
Settings	115200,n,8,2
RThreshold	1

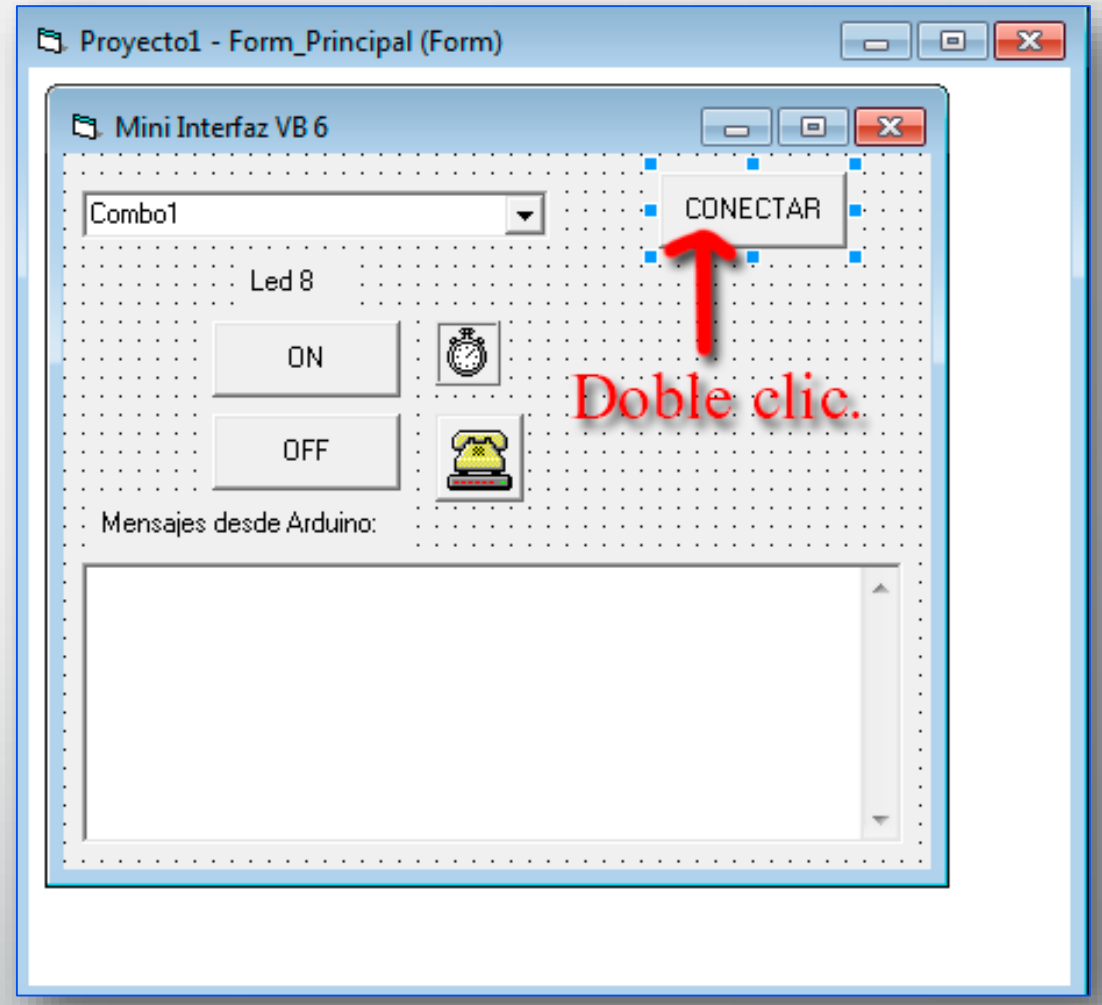
Visual Basic 6

- En la barra de herramientas, “Ver → Código”.



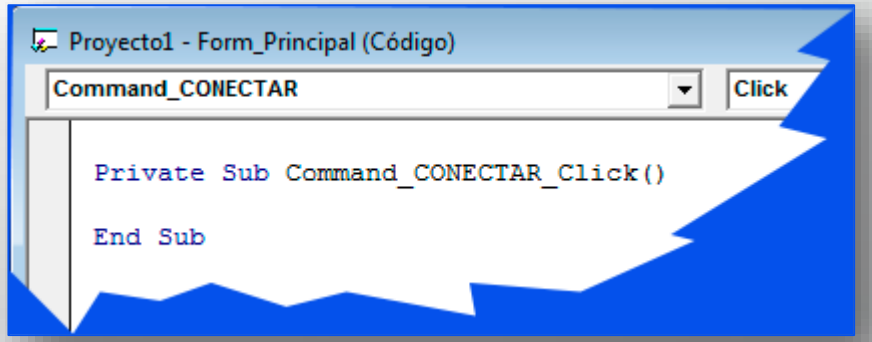
Visual Basic 6

- Pulsamos doble clic en el botón conectar y se nos genera un código.



Visual Basic 6

- Genera este código.



Visual Basic 6

- Introducimos este código.
- Lo que hace es conectar o abrir el puerto serie y si volvemos a pulsar desconectarlo o cerrar el puerto.
- También se activan o desactivan los botones ON y OFF dependiendo si el puerto serie está abierto o no.

Proyecto1 - Form_Principal (Código)

MSComm_Puerto_Serie

```
' Si ya has seleccionado un puerto en el comboBox
' Si pulsas conectar abre el puerto y si no lo cierra.
Private Sub Command_CONECTAR_Click()
    If Not MSComm_Puerto_Serie.PortOpen Then
     MSComm_Puerto_Serie.CommPort = ComboBoxCOM.ItemData(ComboBoxCOM.ListIndex)
       MSComm_Puerto_Serie.PortOpen = True ' Abrir puerto serie.
       Command_CONECTAR.Caption = "DESCONECTAR" ' Mostrar texto en el botón.
       ComboBoxCOM.Enabled = False
       Command_Led_8_ON.Enabled = True ' Activar botón ON.
       Command_Led_8_OFF.Enabled = True ' Activar botón OFF.
    Else
       MSComm_Puerto_Serie.PortOpen = False ' Cerrar puerto serie.
       Command_CONECTAR.Caption = "CONECTAR" ' Mostrar texto en el botón.
       ComboBoxCOM.Enabled = True
       Command_Led_8_ON.Enabled = False ' Desactivar botón ON.
       Command_Led_8_OFF.Enabled = False ' Desactivar botón OFF.
    End If
End Sub
```

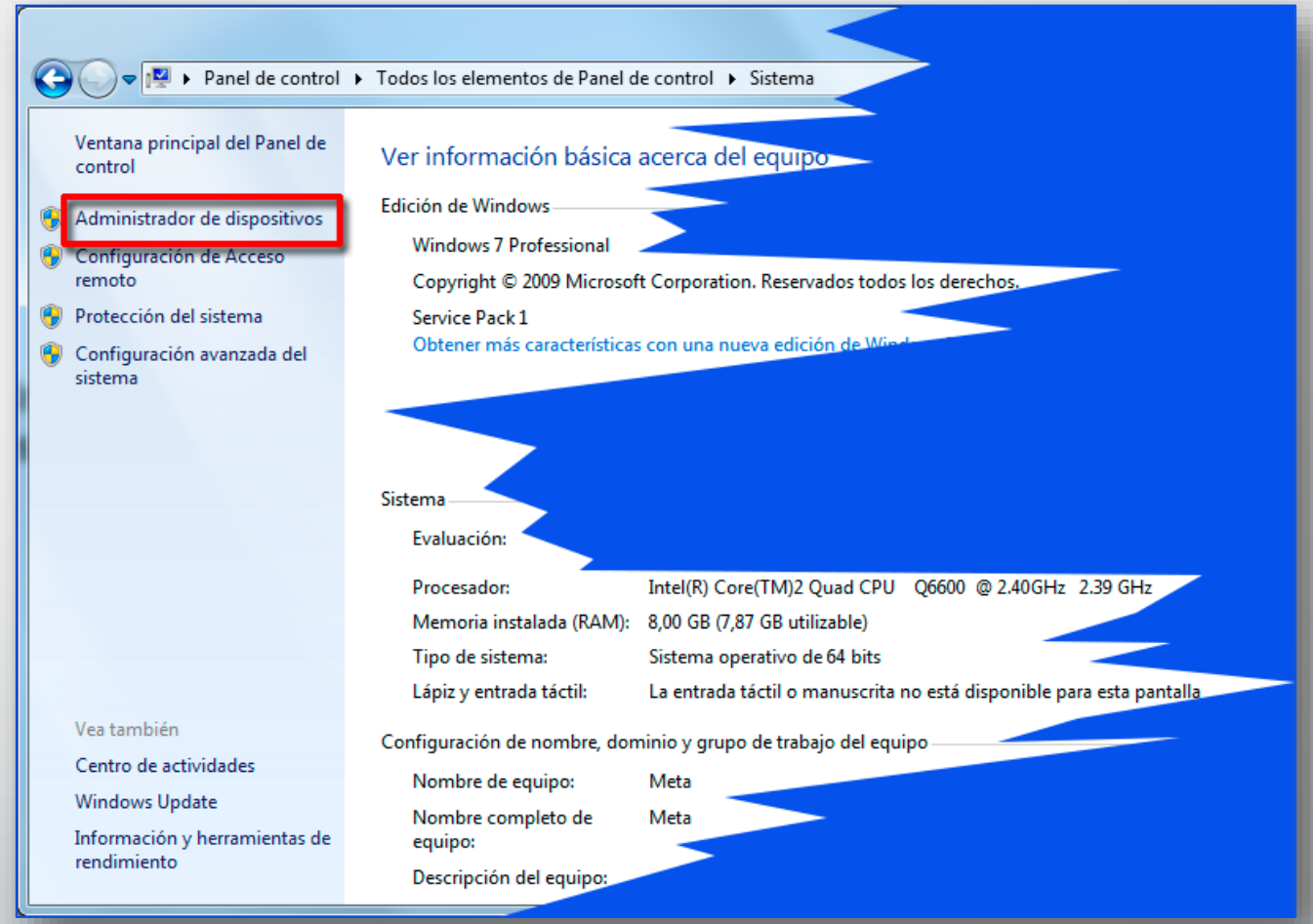
Visual Basic 6

```
Private Sub ComboBoxCOM_Click()
    ' Configurar MSComm
    With Me.ComboBoxCOM
        MSComm_Puerto_Serie.CommPort = ComboBoxCOM.ItemData(ComboBoxCOM.ListIndex)
    End With
    Debug.Print "Puerto seleccionado: COM" & MSComm_Puerto_Serie.CommPort
    'MsgBox "Puerto seleccionado: COM" & MSComm_Puerto_Serie.CommPort
End Sub
```

- Añadimos en la parte más alta del editor de código de VB 6 este trozo de código.
- Es para seleccionar el puerto COM deseado.

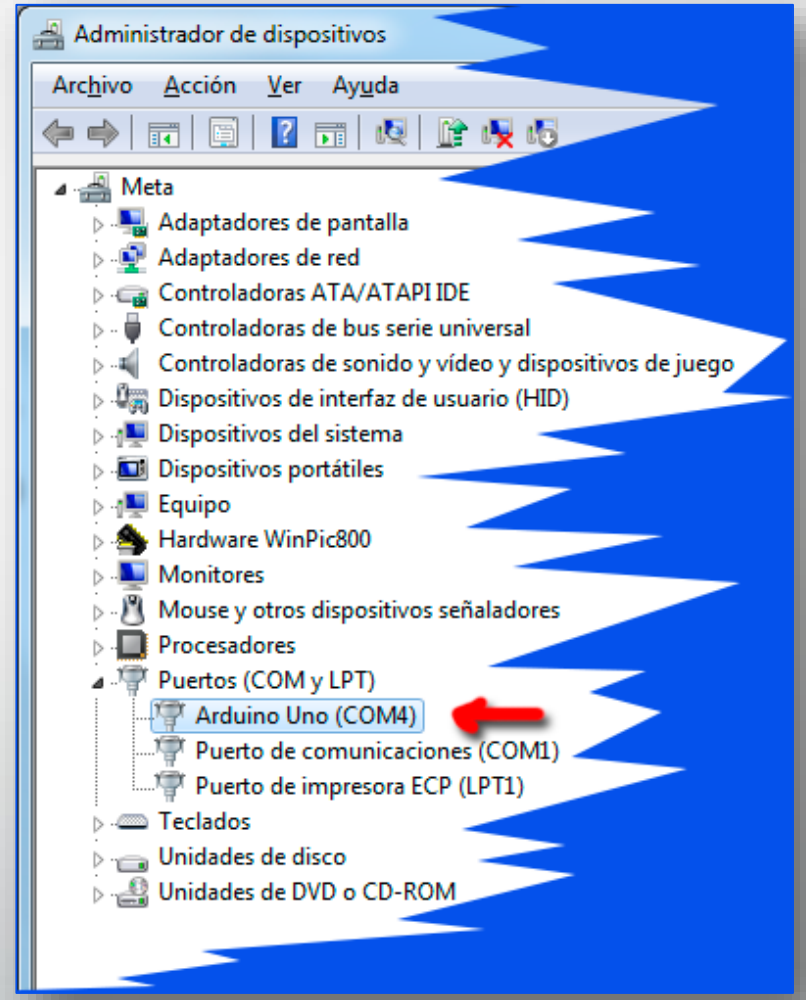
Visual Basic 6

- Para saber que puerto estamos usando con Arduino.
- Entras desde Windows "Inicio → Panel de control\Todos los elementos de Panel de control\Sistema".
- Luego pulsas "Administrador de dispositivos".



Visual Basic 6

- En mi caso, se usa y usaremos el puerto **COM4**.
- Sabiendo el puerto a usar, ya podemos cerrar la ventana "Administrador de dispositivos".



Visual Basic 6

- Haz doble clic en el primer botón como muestra en la imagen.

Visual Basic 6

- Al hacer doble clic en el primer botón, genera unos códigos automáticamente.
- Dentro de ella vamos a escribir los comandos que entiende Arduino IDE que hemos programado.

```
Private Sub Command_Led_8_ON_Click()

End Sub
```

Visual Basic 6

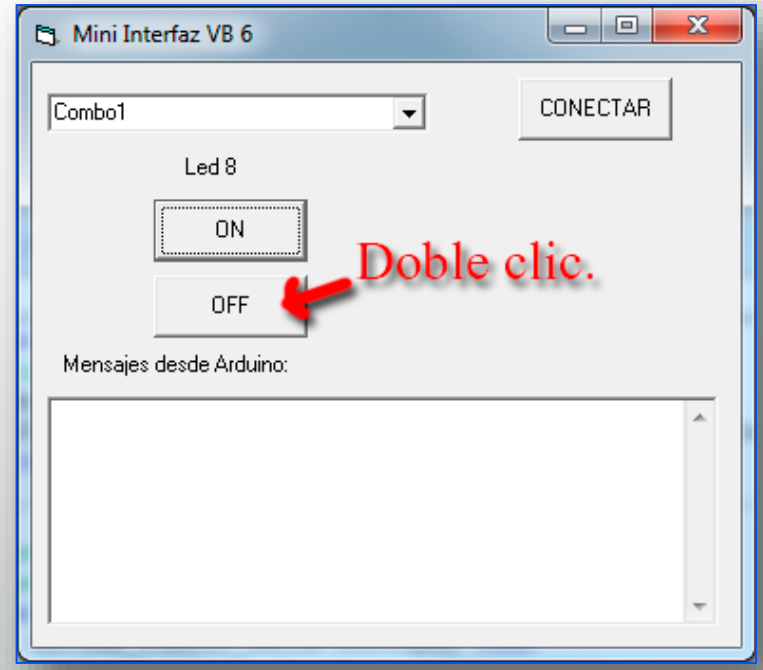
Recordar que los comandos que hemos programado para el Led del pin 8 se llama **Led_8_ON** para encender y para apagar **Led_8_OFF**.

Este es el trozo de código a insertar para el primer botón para encender el Led del pin 8. Aunque lo he llamado Led 8.

```
Private Sub Command_Led_8_ON_Click()
' Envía este cadena de carácter por el puerto serie.
    MSComm_Puerto_Serie.Output = "Led_8_ON"
End Sub
```

Visual Basic 6

- Ahora nos toca hacer doble clic en el segundo botón para poder apagar el Led del pin 8 de Arduino.
- Lo podemos llamar Led 8.



Visual Basic 6

Ahora añadimos el mismo código para el segundo botón con el comando **Led_8_OFF,** así podremos apagar el Led.

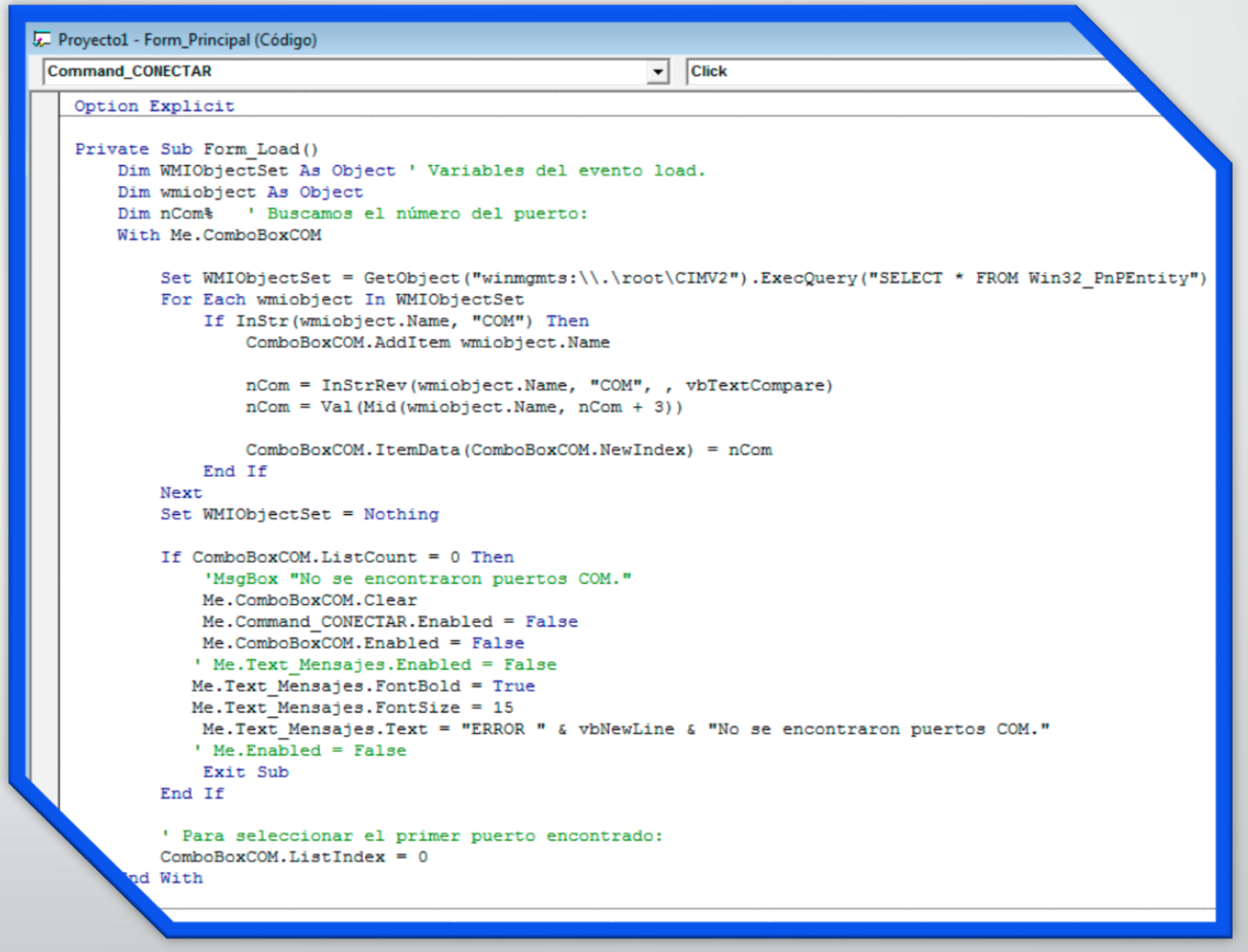
```
Private Sub Command_Led_8_OFF_Click()
' Envía este cadena de carácter por el puerto serie.
    MSComm_Puerto_Serie.Output = "Led_8_OFF"
End Sub
```

Visual Basic 6

Hacer doble clic dentro del formulario sin tocar ningún componente como botones, cmoboBox, label, solo el formulario.

Cuando lo consigas, aparecerá un código generado un Load, así debes introducir todo este código.

En mi caso lo dejo arriba del todo del editor de código.



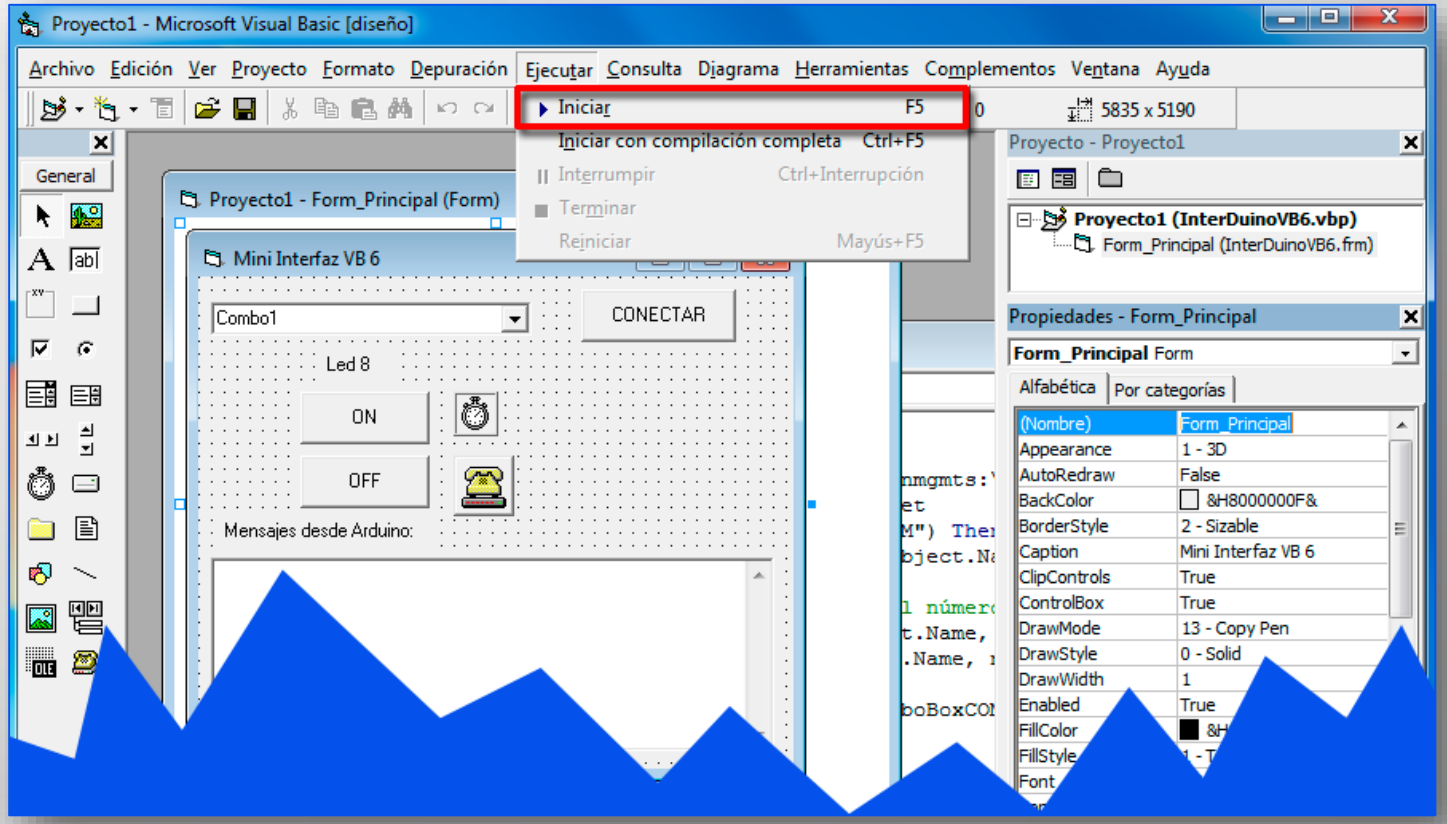

```
Option Explicit

Private Sub Form_Load()
    Dim WMIObjectSet As Object ' Variables del evento load.
    Dim wmiobject As Object
    Dim nCom%    ' Buscamos el número del puerto:
    With Me.ComboBoxCOM

        Set WMIObjectSet = GetObject("winmgmts:\\.\root\CIMV2").ExecQuery("SELECT * FROM Win32_PnPEntity")
        For Each wmiobject In WMIObjectSet
            If InStr(wmiobject.Name, "COM") Then
                ComboBoxCOM.AddItem wmiobject.Name

                nCom = InStrRev(wmiobject.Name, "COM", , vbTextCompare)
                nCom = Val(Mid(wmiobject.Name, nCom + 3))

                ComboBoxCOM.ItemData(ComboBoxCOM.NewIndex) = nCom
            End If
        Next
        Set WMIObjectSet = Nothing

        If ComboBoxCOM.ListCount = 0 Then
            'MsgBox "No se encontraron puertos COM."
            Me.ComboBoxCOM.Clear
            Me.Command_CONECTAR.Enabled = False
            Me.ComboBoxCOM.Enabled = False
           ' Me.Text_Mensajes.Enabled = False
           Me.Text_Mensajes.FontBold = True
           Me.Text_Mensajes.FontSize = 15
            Me.Text_Mensajes.Text = "ERROR " & vbNewLine & "No se encontraron puertos COM."
           ' Me.Enabled = False
            Exit Sub
        End If

        ' Para seleccionar el primer puerto encontrado:
        ComboBoxCOM.ListIndex = 0
    End With
```

Visual Basic 6

- Cambiaremos un poco el código para probarlo ahora mismo.
- Antes llamábamos los comandos **Led_ON** y **Led_OFF**. Ahora lo llamaremos **Led_8_ON** y **Led_8_OFF**.
- Comprobar que estos comandos es capaz de encender y apagar el Led 8 con el "Monitor Serie".

Visual Basic 6

- Cerramos Arduino IDE para dejar el puerto COM4 libre.
- Podemos directamente pulsar F5 para ejecutar la aplicación o nuestro programa.

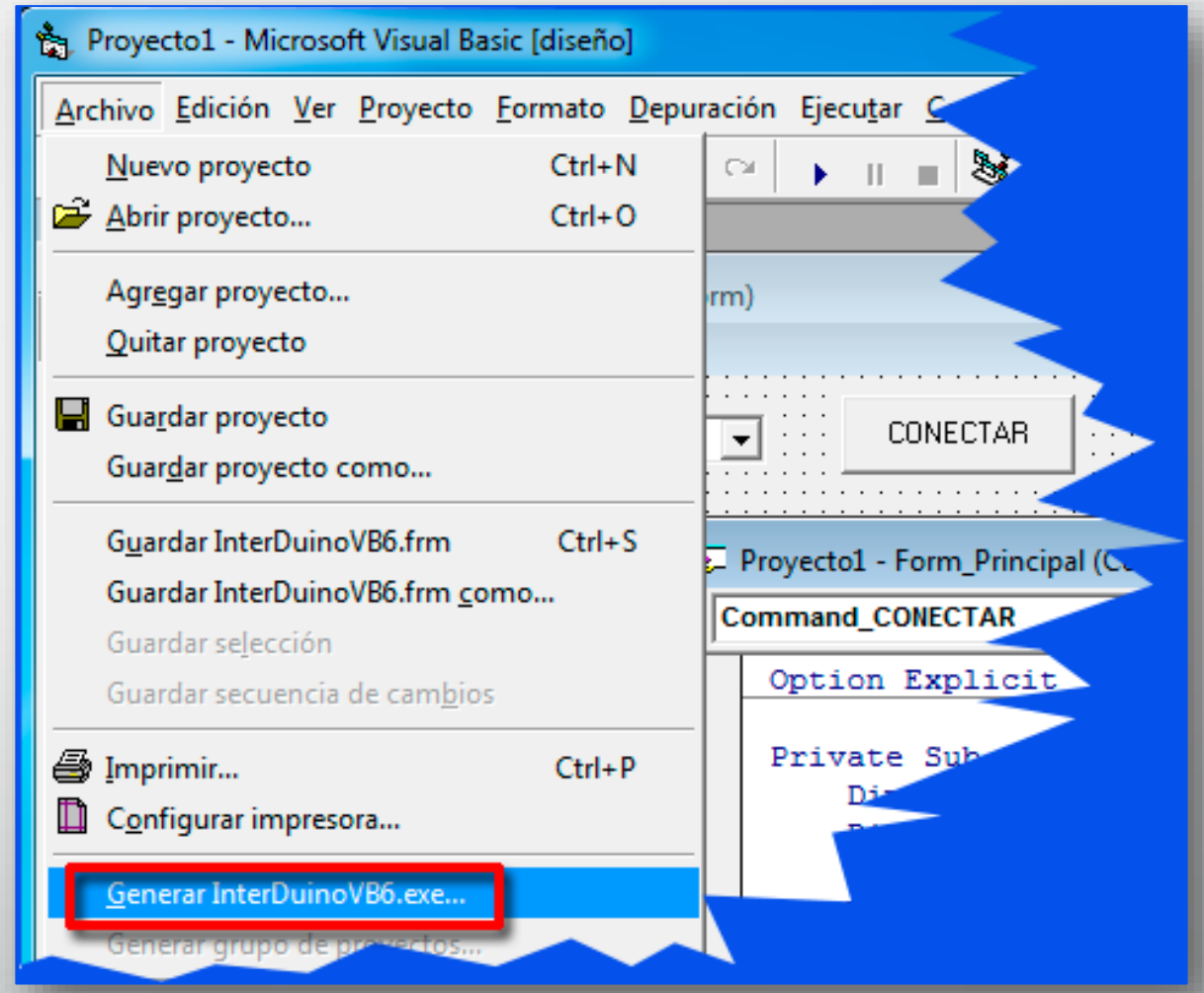


Visual Basic 6

- Hasta aquí es el final sobre el control del puerto serie con Arduino.
- Podemos ver y seleccionar el puerto serie con el comboBox, abrir el puerto y enviar órdenes a Arduino.
- No recibiremos mensajes desde Arduino porque aún no lo hemos programado en este momento.

Visual Basic 6

- Vamos a generar nuestro interfaz con el archivo .exe.
- "Archivo" → "Generar InterDuinoVB6.exe".
- Ya tenemos nuestra aplicación generada.



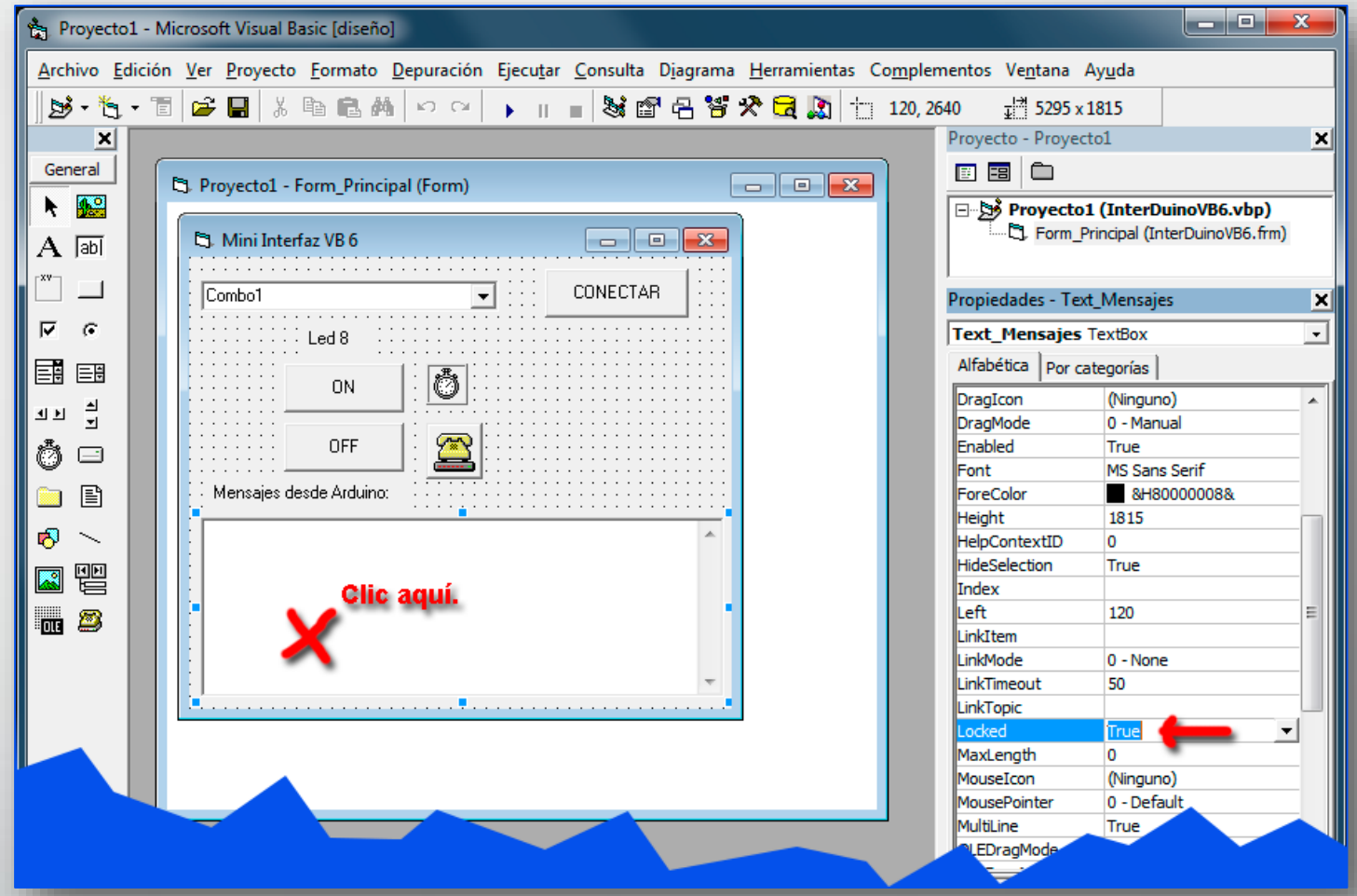
Visual Basic 6

- El código completo por hasta donde hemos llegado con Visual Basic 6.

- Puedes ver el ejemplo **InterDuinoVB6_1** en el directorio de la descarga, InterDuino GENERAL\Windows\VB6\InterDuinoVB6_1

Visual Basic 6

- Aquí nos centraremos en que recibas mensajes y poder verlo en el Text.
- Siguimos adelante para programar de una vez mensajes recibidos desde Arduino.
- En la entrada de mensajes podemos ponerlo en modo lectura y no se puede modificar, si nos interesa ponle True.
- Si no queremos dejarlo en modo lectura, lo dejamos en False.



Visual Basic 6

Propiedades

- En mi caso he dejado esta configuración en las propiedades.

Propiedad	Cambie a
ReadOnly	False
(Name)	Text_Mensajes

Visual Basic 6

- Añadir estos siguientes códigos.
- Estos códigos es para recibir datos del puerto serie desde Arduino y mostrarlo en el Text_Mensajes.
- A continuación, en la página siguiente se muestra el código completo por si te pierdes.

```
Private Sub MSComm_Puerto_Serie_OnComm()
    ' Variable para almacenar datos.
    Dim Dato_Recibido As String
    ' Los datos recibidos lo almacena en la variable.
    Dato_Recibido = MSComm_Puerto_Serie.Input
    ' Los datos de la varible lo muestra en el Text.
    Text_Mensajes.Text = Text_Mensajes.Text & Dato_Recibido
    ' Mantiene el scroll en la entrada de cada mensaje.
    Text_Mensajes.SelStart = Len(Text_Mensajes.Text)
nd Sub
```

Visual Basic 6

- Añadir estos siguientes códigos.
- Estos códigos es para recibir datos del puerto serie desde Arduino y mostrarlo en el Text_Mensajes.
- A continuación, en la página siguiente se muestra el código completo por si te pierdes.

Código desde el principio.

yecto1 - Form_Principal (Código)

Command_Led_8_OFF | Click

```
Option Explicit

Private Sub Form_Load()
    Dim WMIObjectSet As Object ' Variables del evento load.
    Dim wmiobject As Object
    Dim nCom%    ' Buscamos el número del puerto:
    With Me.ComboBoxCOM

        Set WMIObjectSet = GetObject("winmgmts:\\.\root\CIMV2").ExecQuery("SELECT * FROM Win32_PnPEntity")
        For Each wmiobject In WMIObjectSet
            If InStr(wmiobject.Name, "COM") Then
                ComboBoxCOM.AddItem wmiobject.Name

                nCom = InStrRev(wmiobject.Name, "COM", , vbTextCompare)
                nCom = Val(Mid(wmiobject.Name, nCom + 3))

                ComboBoxCOM.ItemData(ComboBoxCOM.NewIndex) = nCom
            End If
        Next
        Set WMIObjectSet = Nothing

        If ComboBoxCOM.ListCount = 0 Then
            'MsgBox "No se encontraron puertos COM."
            Me.ComboBoxCOM.Clear
            Me.Command_CONECTAR.Enabled = False
            Me.ComboBoxCOM.Enabled = False
           ' Me.Text_Mensajes.Enabled = False
           Me.Text_Mensajes.FontBold = True
           Me.Text_Mensajes.FontSize = 15
            Me.Text_Mensajes.Text = "ERROR " & vbNewLine & "No se encontraron puertos COM."
           ' Me.Enabled = False
            Exit Sub
        End If

        ' Para seleccionar el primer puerto encontrado:
        ComboBoxCOM.ListIndex = 0
    End With
End Sub
```

```
rivate Sub ComboBoxCOM_Click()
    ' Configurar MSComm
    With Me.ComboBoxCOM
        MSComm_Puerto_Serie.CommPort = ComboBoxCOM.ItemData(ComboBoxCOM.ListIndex)
    End With
    Debug.Print "Puerto seleccionado: COM" & MSComm_Puerto_Serie.CommPort
    'MsgBox "Puerto seleccionado: COM" & MSComm_Puerto_Serie.CommPort
End Sub
```

```
' Si ya has seleccionado un puerto en el comboBox
' Si pulsas conectar abre el puerto y si no lo cierra.
Private Sub Command_CONECTAR_Click()
    If Not MSComm_Puerto_Serie.PortOpen Then
     MSComm_Puerto_Serie.CommPort = ComboBoxCOM.ItemData(ComboBoxCOM.ListIndex)
       MSComm_Puerto_Serie.PortOpen = True ' Abrir puerto serie.
       Command_CONECTAR.Caption = "DESCONECTAR" ' Mostrar texto en el botón.
       ComboBoxCOM.Enabled = False
       Command_Led_8_ON.Enabled = True ' Activar botón ON.
       Command_Led_8_OFF.Enabled = True ' Activar botón OFF.
    Else
       MSComm_Puerto_Serie.PortOpen = False ' Cerrar puerto serie.
       Command_CONECTAR.Caption = "CONECTAR" ' Mostrar texto en el botón.
       ComboBoxCOM.Enabled = True
       Command_Led_8_ON.Enabled = False ' Desactivar botón ON.
       Command_Led_8_OFF.Enabled = False ' Desactivar botón OFF.
    End If
End Sub
```

```
Private Sub Command_Led_8_OFF_Click()
' Envía este cadena de carácter por el puerto serie.
    MSComm_Puerto_Serie.Output = "Led_8_OFF"
End Sub

Private Sub Command_Led_8_ON_Click()
' Envía este cadena de carácter por el puerto serie.
    MSComm_Puerto_Serie.Output = "Led_8_ON"
End Sub


Private Sub MSComm_Puerto_Serie_OnComm()
    ' Variable para almacenar datos.
    Dim Dato_Recibido As String
    ' Los datos recibidos lo almacena en la variable.
    Dato_Recibido = MSComm_Puerto_Serie.Input
    ' Los datos de la varible lo muestra en el Text.
    Text_Mensajes.Text = Text_Mensajes.Text & Dato_Recibido
    ' Mantiene el scroll en la entrada de cada mensaje.
    Text_Mensajes.SelStart = Len(Text_Mensajes.Text)
End Sub
```

Fin de código.

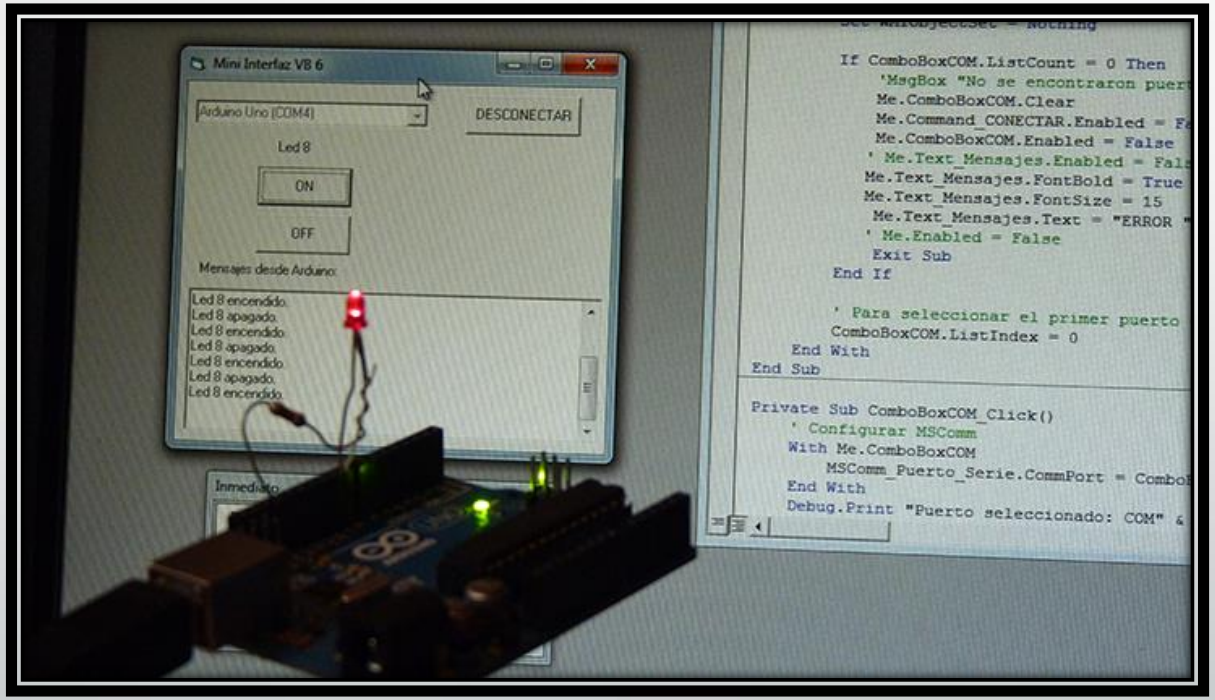
Visual Basic 6

- El código completo por hasta donde hemos llegado con Visual Basic 6.

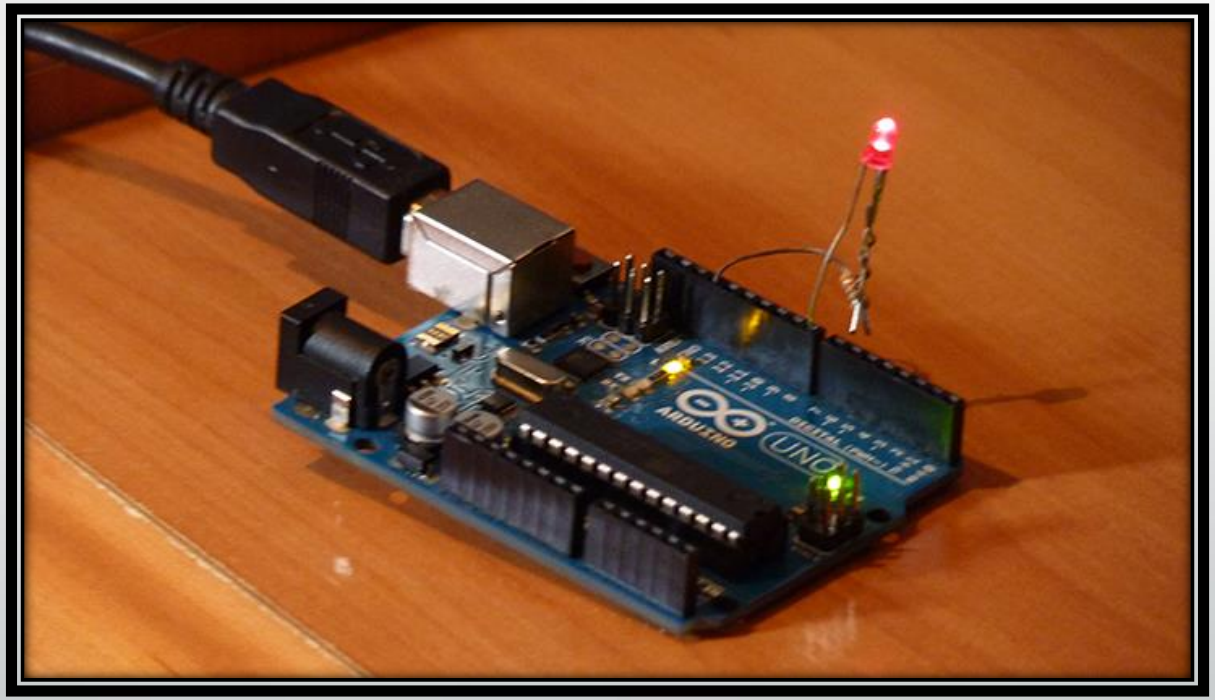
- Puedes ver el ejemplo **InterDuinoVB6_2** en el directorio de la descarga, InterDuino GENERAL\Windows\VB6\InterDuinoVB6_2

Fotos

Fotos


Mini Interfaz VB 6
Arduino Uno (COM4)
DESCONECTAR
Led 8
ON
OFF
Mensajes desde Arduino:
Led 8 encendido.
Led 8 apagado.
Led 8 encendido.
Led 8 apagado.
Led 8 encendido.
Led 8 apagado.
Led 8 encendido.
If ComboBoxCOM.ListCount = 0 Then
'MsgBox "No se encontraron puer
Me.ComboBoxCOM.Clear
Me.Command_CONECTAR.Enabled = F
Me.ComboBoxCOM.Enabled = False
' Me.Text_Mensajes.Enabled = Fal
Me.Text_Mensajes.FontBold = True
Me.Text_Mensajes.FontSize = 15
Me.Text_Mensajes.Text = "ERROR
' Me.Enabled = False
Exit Sub
End If
' Para seleccionar el primer puerto
ComboBoxCOM.ListIndex = 0
End With
End Sub
Private Sub ComboBoxCOM_Click()
' Configurar MSComm
With Me.ComboBoxCOM
MSComm_Puerto_Serie.CommPort = Combo
End With
Debug.Print "Puerto seleccionado: COM" &

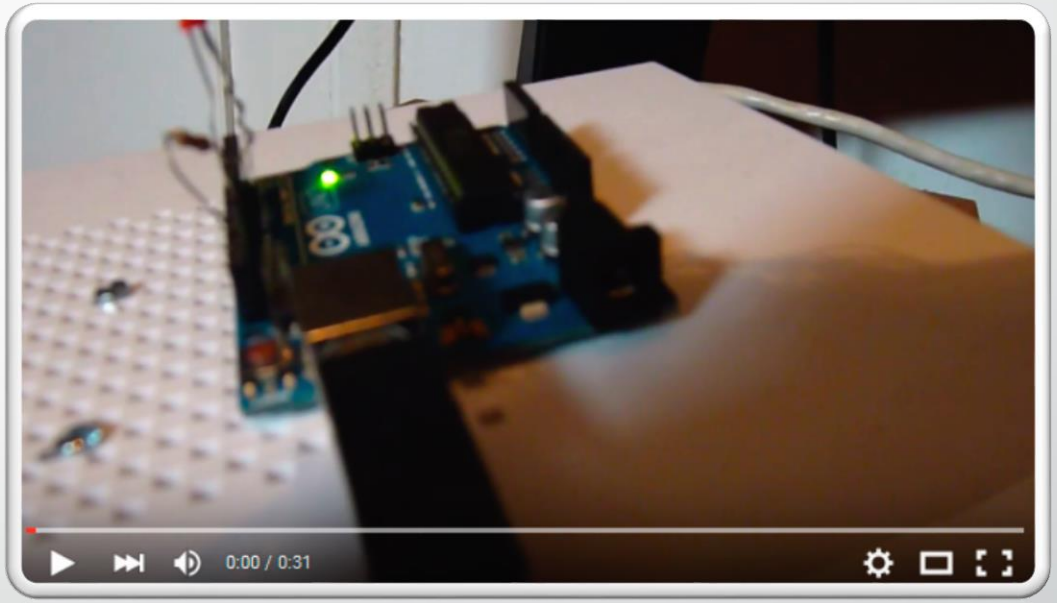
Fotos



Vídeo

Vídeo

https://www.youtube.com/watch?v=SisRHK9_at4



Enlaces de interés

Enlaces de interés

- Foro MSDN: Puedes participar en el foro oficial de Visual Basic 6.
- Arduino: Información oficial sobre el mundo de Arduino y foros.
- Electrónica PIC: Blog oficial del creador de este tutorial.

Versión del tutorial

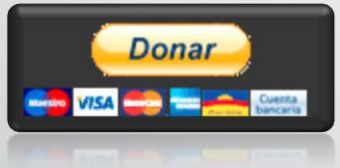
Versión: 1.00

Contacto

Contacto

Contactar: metaconta@gmail.com

PayPal: egiptoman@gmail.com



- Puedes publicar este tutorial en tu Web, foro, blog, CD-ROM, DVD-ROM, Blu-Ray en revistas oficial en papel o cualquier otro medio.

Autor

Autor

Ángel Acaymo M. G.