

Interfaz Gráfica Web con Dataino.it para placas Arduino y compatibles

Abril, J., Chenoweth, I., Meza I., Elizarrarás J., Felix, J.

Resumen—En este trabajo se propone el desarrollo de Interfaces gráficas de usuario con elementos visuales de tipo drag and drop dentro del navegador Google Chrome, usando el complemento Dataino.it, el cual es, en esencia, una plataforma que interactúa en tiempo real a través de una conexión de: USB, LoRa o Ethernet con cualquier placa Arduino o compatible, además facilita al desarrollador de software la generación de código funcional en el diseño de las interfaces.

Palabras clave – Arduino, Dataino.it, Google Chrome, GUI, USB.

INTRODUCCIÓN

La tendencia del internet de las cosas (IoT) [1], implica que cada vez habrá más conexiones a internet ya sean de máquina a máquina (M2M) [2] o de humano a máquina (H2M) [3]. Hoy en día, se cuenta con tabletas, *smarthphones*, computadoras de escritorio entre otros dispositivos, donde las aplicaciones compatibles con los navegadores se han hecho cada vez más populares por la portabilidad. Asimismo, existen múltiples herramientas que permiten generar *Graphic user interfase* (GUI) para diferentes placas electrónicas de desarrollo, pero la facilidad de tener el desarrollo dentro del navegador se convierte en una tendencia, por tener el beneficio de la portabilidad en esos dispositivos.

Dataino.it es una herramienta Web muy intuitiva para la generación de GUI avanzadas con componentes visuales y generación de código funcional para la comunicación con placas Arduino y compatibles. Incluye todo lo necesario para establecer las conexiones y comunicarse usando las siguientes opciones: Network (red), *Universal Serial Bus* (USB) y LoRa, con el fin de operar componentes. Para la elaboración del diseño de las GUI, Dataino.it contiene elementos gráficos de tipo drag and drop que operan en tiempo real.

- Push
- Switch
- Slider
- Time Slider
- Text
- Led
- Progress Bar
- Scale
- Scale Value
- LCD
- Circle
- Vu meter
- Touch
- Charge Meter
- Crono
- GPS

Un ejemplo con los elementos mencionados anteriormente se pueden visualizar en la fig. 1.

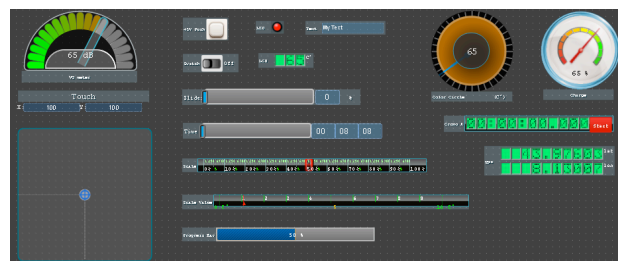


Figura 1. Elementos gráficos de Dataino.it

Como se puede observar, hay una gran variedad de elementos visuales que podemos utilizar para desarrollar GUI complejas que interactúen con Arduino tanto para enviar datos a la placa y mostrar datos enviados por la placa. El código generado por Dataino.it se copia y se sube a la placa con el *software* de Arduino, lo que lo convierte en una aplicación funcional para operar en tiempo real. Es posible que el código sirva de plantilla para elaborar proyectos más complejos dentro de Arduino, y que, con este escenario, se pueda interactuar con componentes electrónicos conectados a su placa.

DESARROLLO

En este trabajo se propone el desarrollo de una GUI utilizando la herramienta Dataino.it que permita interactuar en tiempo real desde el navegador Google Chrome en cualquier dispositivo con la placa electrónica Arduino. Ésta se conecta a una placa de 8 diodos emisores de luz (LEDs) en las salidas y corresponden con 8 botones visuales en la GUI. Fig. 2. muestra el diagrama propuesto de este proyecto, involucrando los elementos principales.

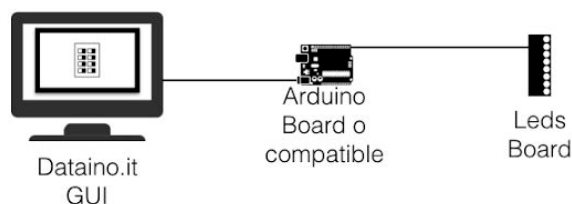


Figura 2. Diagrama de proyecto.

El software utilizado es el siguiente:

- Google Chrome Versión 51.0.2704.106 m.
- Dataino.it Beta 1.5 [4].
- Extensión de Google Chrome Arduino to Dataino Serial Proxy Versión 1.0.3 [5].

- Arduino 1.6.9 [6].

A continuación se enlista el hardware:

- Computadora Intel(R) Atom(TM) CPU N570 @ 1.66 GHz 1.67 GHz RAM: 2.00 GB.
- Cable USB a micro USB.
- Placa RobotDyn uno R3 [7].
- Cables dupont macho a macho.
- Placa de LEDs [8].
- Protoboard.

El primer paso consiste en la instalación del navegador Google Chrome. Es importante cuidar y revisar que las versiones de las herramientas sean compatibles, se recomienda verificar compatibilidad en la ayuda de cada herramienta. Una vez realizada la instalación, se deberá ingresar al sitio web de Dataino.it, y entonces, se crea una cuenta de usuario, se recomienda que el correo solicitado por la aplicación sea de gmail. Después de registrarse se accede al sitio, para que Dataino.it pueda conectarse a la placa Arduino, requiere de la instalación de un complemento, el cual se descarga del mismo sitio, este se puede observar en la Fig. 3.

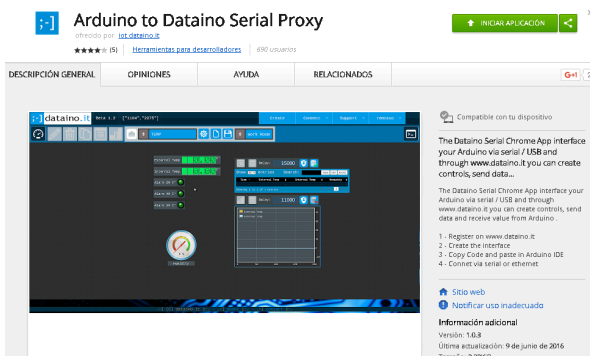


Figura 3. Complemento para conexión.

Se puede detectar que se trata de una aplicación para generar GUI avanzadas que permiten la interacción con la placa de desarrollo. Podemos administrar nuestros proyectos, generar código y comunicarnos con la placa de desarrollo. Creamos un nuevo proyecto, en este caso sólo se colocarán algunos *switches* que permitan encender y apagar LEDs, ver Fig. 4.

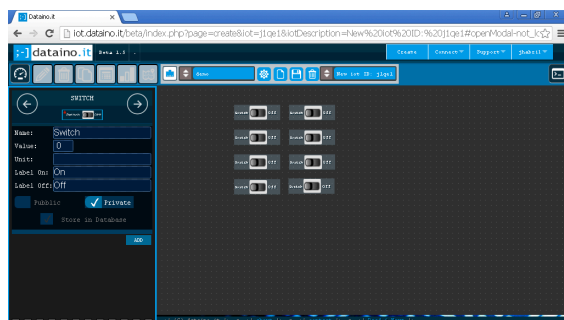


Figura 4. Botones y propiedades.

Una vez terminado el diseño del GUI, Dataino.it genera

automáticamente todo el código necesario para ejecutar la aplicación (Fig. 5). Éste debe ser copiado y pegado en el software de Arduino para subir a la placa, cabe señalar que el mismo proceso se puede llevar a cabo en otras placas que tienen el mismo funcionamiento que Arduino y que, a su vez, son compatibles como fue nuestro caso utilizando la placa RobotDyn uno R3. Es importante resaltar que esta versión web de Dataino.it no se puede conectar directamente a la placa para hacer la descarga del código.

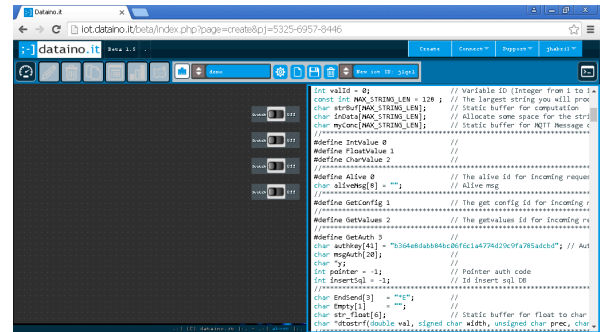


Figura 5. Generador de código.

Una vez descargado el código en el software de Arduino, es necesario hacer modificaciones para ajustar la disposición de las entradas y salidas de propósito general (GPIO), al código generado. Los métodos modificados fueron: *setup()* y *loop()*; los cuales se describen a continuación:

- En el método *setup()*, se modificó el siguiente código:

```
void setup() {
  // Código modificado manualmente
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(12,OUTPUT);
}
```

- En el método *loop()*, se modifica el siguiente código:

```
void loop() {
  // Código modificado manualmente
  digitalWrite(2,SW00);
  digitalWrite(3,SW01);
  digitalWrite(5,SW02);
  digitalWrite(6,SW03);
  digitalWrite(7,SW04);
  digitalWrite(8,SW05);
  digitalWrite(10,SW06);
  digitalWrite(12,SW07);
}
```

Después de realizar las modificaciones necesarias se procede a subir el código a la placa. Enseguida, se debe conectar la placa. Dataino.it cuenta con tres opciones para la comunicación directa con las placas:

- Conexión USB: Es un bus estándar industrial muy conocido y utilizado, define los cables, protocolos, y conectores usados para conectar, comunicar y proveer de alimentación eléctrica entre periféricos, computadoras y dispositivos electrónicos. A diferencia de otras conexiones, USB no necesita

ningún tipo de configuración adicional.

- Conexión de red: Dataino.it utiliza el protocolo de conectividad de máquina a máquina MQTT [9] conocido como el Internet de las cosas IoT, con el uso de WebSocket, que es una tecnología que proporciona un canal de comunicación bidireccional sobre un único socket TCP. Está diseñada para ser implementada en navegadores modernos y servidores web, pero puede utilizarse por cualquier aplicación en una red cliente/servidor, mediante una red Ethernet.
- Conexión LoRa [10]: Es un protocolo que permite hacer conexiones de tipo LPWA [11] a más de 10Km desarrollado por la Alianza LoRa® la cual es una asociación sin fines de lucro enfocada a la IoT.

En este caso en particular y debido a las herramientas disponibles se utilizará la conexión USB. A continuación, se procede a enlazar nuestra GUI con la placa, desde la página de Dataino.it, en modo de diseño, en el menú *Connect* se selecciona la opción Serial, en esta área, se elige el puerto USB donde está conectada la placa (COM1, COM2, ... COMn) y el rango de bits (Bit Rate) para nuestra aplicación es de 57600, se presiona el botón "Load", si todo está correcto, ya se puede comunicar directamente con la placa RobotDyn uno R3, como se ve en la Fig. 6.



Figura 6. Encendido de LEDs.

En esta aplicación en particular, se enciende y apaga los LEDs a través de imágenes en el GUI que representan *switches*. Con esto queda demostrado que se ha logrado desarrollar la aplicación que puede manipular dispositivos a través de una placa Arduino o compatible.

CONCLUSIONES

Se tiene una guía práctica para el desarrollo de una GUI que pueda manipular mediante una placa Arduino otro dispositivo electrónico. Para proyectos futuros se sugiere el uso de componentes electrónicos más complejos o programas que requieren procesos más elaborados. Este tipo de práctica propicia el desarrollo de proyectos enfocados a IoT, que permitan la manipulación de dispositivos electrónicos de

manera remota.

APENDICES

Muestras del código generado por Dataino.it de nuestra aplicación.

```

//*****
/** created 2014
/** by dataino www.dataino.it
/** modified 18 Aug 2014
/** by G. Carmelo
//*****
/** Dataino Serial version 001
/** Copyright (c) 2015 Carmelo G.
/** If you require a license, see
/** http://www.opensource.org/licenses/bsd-
/** license.php
//*****
/** PROGRAMM VARIABLE AND CONST
//*****
boolean VAR_IN = false;
// switch fon incoming variable
boolean VAL_IN = false;
// switch fon incoming value
int valId = 0;
// Variable ID
boolean stringComplete = false;
unsigned long Timer = 100;
// Time basis to do something
const int MAX_STRING_LEN = 256;
// the largest string you will process
char stringBuffer[MAX_STRING_LEN];
// static buffer for computation
char inData[MAX_STRING_LEN];
// Allocate some space for the string
char inChar = -1;
// Where to store the character read
byte index = 0;
// Index into array for store the char
char IOT[6] = "j1qel";
char Live[5] = "Live";
char GetConfig[5] = "GetC";
char GetValues[5] = "GetV";
char SetVar[5] = "SetV";
char EndSend[3] = "E";
//=====
// PROJECT ID:
//=====
// INCOMING OUTGOING GAUGE VARIABLES SETTING
//=====
//
//
//=====
// SEND VALUE TO MONITOR
//=====
void SendValue(int SendVar)
{
//*****
/** SEND VALUE TO MONITOR:
//*****
char OutBuffer[MAX_STRING_LEN];
// static buffer for computation
sprintf( OutBuffer, "V;%s;%i;", IOT, SendVar );
Serial.print( OutBuffer );
switch ( SendVar )
{
case SW04id:
// SEND VALUE "Switch >> SW04" TO MONITOR
Serial.print( SW04 );
break;
case SW03id:
// SEND VALUE "Switch >> SW03" TO MONITOR
Serial.print( SW03 );

```

```

    break;
case SW07iD:
    // SEND VALUE "Switch >> SW07" TO MONITOR
    Serial.print( SW07 );
    break;
case SW06iD:
    // SEND VALUE "Switch >> SW06" TO MONITOR
    Serial.print( SW06 );
    break;
case SW01iD:
    // SEND VALUE "Switch >> SW01" TO MONITOR
    Serial.print( SW01 );
    break;
case SW02iD:
    // SEND VALUE "Switch >> SW02" TO MONITOR
    Serial.print( SW02 );
    break;
case SW00iD:
    // SEND VALUE "Switch >> SW00" TO MONITOR
    Serial.print( SW00 );
    break;
case SW05iD:
    // SEND VALUE "Switch >> SW05" TO MONITOR
    Serial.print( SW05 );
    break;
}
Serial.println( F( "|" ) );
}
. . .

```

REFERENCIAS

- [1] Michael Weyrich and Christof Ebert, "Reference Architectures for the Internet of Things", IEEE Software (Volume: 33, Issue: 1), pp. 112-116, 2016.
- [2] Jaewoo Kim, Jaiyong Lee, Jaeho Kim and Jaeseok Yun "M2M Service Platforms: Survey, Issues, and Enabling Technologies", IEEE IEEE Communications Surveys & Tutorials (Volume: 16, Issue: 1), pp. 61-76, 2014.
- [3] Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", Wiley, Canada, 2013.
- [4] <http://iot.dataino.it/>
- [5] https://chrome.google.com/webstore/detail/arduino-to-dataino-serial/aacandifkeejacfdplimpicafjjkbbga?utm_source=chrome-app-launcher-info-dialog
- [6] <https://www.arduino.cc/en/Main/Software>
- [7] http://robotdyn.com/catalog/boards/uno_r3_ch340g_compatible_with_arduino_uno_r3/ <http://www.tropheus.demon.co.uk/nplan.htm>
- [8] <http://hotboards.mitiendanube.com/productos/leds/>
- [9] <http://mqtt.org/>
- [10] <https://www.lora-alliance.org/>
- [11] <https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRa-Alliance-Whitepaper-LPWA-Technologies.pdf>

AUTORES

Abril García José Humberto. abril@uthermosillo.edu.mx
 Profesor de Tiempo Completo de la Universidad Tecnológica de Hermosillo.

Meza Ibarra Iván Dostoyewski. imeza@uthermosillo.edu.mx
 Profesor de Tiempo Completo de la Universidad Tecnológica de Hermosillo.

Chenoweth Chenoweth Ivan Rogelio.
ichenoweth@uthermosillo.edu.mx Profesor de Asignatura de la Universidad Tecnológica de Hermosillo.

Elizarrarás Quiroz José de Jesús.
jelizarraras@uthermosillo.edu.mx Profesor de Asignatura de la Universidad Tecnológica de Hermosillo.

Félix Callejas Judith Cristina jcristina.felix@uttijuana.edu.mx
 Profesor de Tiempo Completo de la Universidad Tecnológica de Tijuana.