

Representación gráfica de datos recolectados de una fotocelda a través de Arduino y Processing

Graphical representation of data collected from a photocell through Arduino and Processing

Javier Díaz Sánchez

Benemérita Universidad Autónoma de Puebla, México

jdiazsz@hotmail.com

Resumen

La educación sustentada en un modelo constructivista debe adaptar sus contenidos a la realidad que se vive, debe ser capaz de integrar a su comunidad en un entorno de progreso sin soslayar la relación entre lo técnico y lo humano; es una integración de contenidos que atiende a una realidad cultural nacional y global. En nuestro país la necesidad de “generar ciencia” se ha visto evidenciada ante los nuevos embates geopolíticos que hoy día impactan económicamente, y que impiden la proliferación de empresas transnacionales que reducen de manera importante el empleo; ante esta situación, ya no es suficiente enseñar a manipular maquinas o sistemas, sino a consolidar el desarrollo de tecnologías a partir de las diversas teorías y herramientas actuales que proporciona la globalización. Este enfoque se encuentra implícito en el Modelo Educativo y su actual Reforma (RIEMS); gestionada en sus los ejes estructurales, que cumplen con las directrices de formativas denominadas competencias; estas estructuras que desarrollan alcances, formas y contenidos; mismos que son verificables a través de los instrumentos cuantitativos y cualitativos adecuados. Por ende, a partir de lo anterior, este proyecto expone sus metas y delimitaciones en un marco referencial acorde al sentido del modelo y su espacio de desarrollo (Nivel Medio Superior) como una propuesta de práctica de laboratorio para la asignatura de Informática del Modelo Universitario Minerva (MUM), correspondiente al tercer grado del Bloque III, teniendo como propósito el desarrollo teórico y do

cumental de una interface básica de interpretación de datos obtenidos de un sensor conectado a una Placa Arduino, y el para su procesamiento gráfico de se hace uso del lenguaje Processing.

Para apoyarse en el desarrollo sostenible de dicha práctica, se hace necesario de la transversalidad entre las áreas de Informática y el conocimiento de la electrónica básica propia de la asignatura de física, lo que representa una oportunidad rica en aprendizajes favorables al modelo educativo. Ahora bien, el desarrollo de cualquier práctica que involucre objetos de aprendizaje en los que intervienen elementos electrónicos o químicos implican un compromiso importante en su manejo, ya que aunque teóricamente en el desarrollo se expone un ambiente controlable; no se debe omitir que la realidad exige una responsabilidad mayor para el profesorado y el alumnado, situación que invita a considerar el uso de equipos específicos para la práctica.

Finalmente, el resultado a la aplicación de esta propuesta, expone nuevas realidades y necesidades educativas, tecnológicas y de emprendimiento que como docentes debemos afrontar, si realmente se desea aportar un cambio en el futuro del alumnado, así como la generación de ambientes de aprendizaje significativos que aporten al interés científico y tecnológico.

Palabras Clave: Arduino, Processing, Informática, Competencias, NMS, Digital, Electrónica, fotoresistor, fotocelda.

Abstract

Education based on a constructivist model must adapt its contents to the reality that is lived, must be able to integrate its community in an environment of progress without overlooking the relationship between the technical and the human; is an integration of content that attends to a national and global cultural reality. In our country, the need to "generate science" has been evidenced by the new geopolitical clashes that today impact economically and that prevent the proliferation of transnational companies that significantly reduce employment; In this situation, it is no longer enough to teach to manipulate machines or systems, but to consolidate the development of technologies based on the various theories and current tools that globalization provides. This approach is implicit in the Educational Model and its current Reform (RIEMS); managed in its structural axes, which comply with training guidelines

called competencies; these structures that develop reaches, forms and contents; which are verifiable through appropriate quantitative and qualitative instruments. Therefore, from the above, this project sets out its goals and delimitations in a referential framework according to the meaning of the model and its development space (Higher Middle Level) as a proposal of laboratory practice for the subject of Computer Model University Minerva (MUM), corresponding to the third degree of Block III, aiming at the theoretical and documentary development of a basic interface of interpretation of data obtained from a sensor connected to an Arduino, and for its graphic processing of use of the Language Processing.

To support the sustainable development of this practice, it is necessary to cross the areas of Informatics and knowledge of the basic electronics of the subject of physics, which represents a rich opportunity in learning favorable to the educational model. However, the development of any practice that involves learning objects involving electronic or chemical elements implies an important commitment in its management, since although theoretically in development a controllable environment is exposed; it should not be omitted that reality demands greater responsibility for teachers and students, a situation that invites us to consider the use of specific equipment for practice.

Finally, the result to the application of this proposal, exposes new realities and educational, technological and entrepreneurial needs that as teachers we must face, if we really want to bring about a change in the future of students, as well as the generation of significant learning environments that contribute to the scientific and technological interest.

Key words: Arduino, Processing, Computing, Competences, NMS, Digital, Electronics, photoresistor, photocell.

Fecha Recepción: Febrero 2017

Fecha Aceptación: Julio 2017

Introducción

Constructivismo

La teoría constructivista se sustenta en la necesidad de construir el conocimiento a través de hechos desarrollados en contextos ricos de aprendizaje significativo, además de “proponer que el ambiente de aprendizaje debe sostener múltiples perspectivas o interpretaciones de la realidad, construcción de conocimiento, actividades basadas en experiencias ricas en contexto” (Jonasse, 1991). Todo centrado en la construcción a través de tareas auténticas por su relevancia y utilidad en el mundo real y no sólo en su reproducción.

Sostiene que el ser humano tanto en los aspectos cognoscitivos y sociales del comportamiento, como en los afectivos, no es un simple producto del ambiente ni un escueto resultado de sus inherentes destrezas, sino una construcción propia que se va cimentando día a día como producto de la interacción entre esos dos factores; afirma que el conocimiento no es una copia de la realidad, sino una construcción propia del ser y del ambiente que lo rodea como una relación biunívoca entre estas entidades.

Los autores Savery y Duffy (1996), describen la posición constructivista en tres puntos de manera general:

- a) El aprendizaje sucede siempre como resultado de nuestras interacciones con el contexto.*
- b) El estímulo para aprender viene de un conflicto cognitivo interno y personal.*
- c) El conocimiento se genera socialmente, a través de poner a prueba nuestras propias representaciones con las de los demás.*

Por otro lado, aunque el constructivismo ha sido uno de los pilares más importantes en la formación de nuevos paradigmas educativos, debe entenderse como un marco epistemológico de referencia, no como un modelo de aprendizaje o de enseñanza (Savery y Duffy, 1996; Gómez-Granell, C., & Coll, 1994; Millán, 1995; etc.).

Para exhibir un panorama de este marco de referencia, es importante mencionar las características que lo consolidan; por ende, con el Constructivismo.

- Se aprende construyendo esquemas mentales.
- Se desarrollan aprendizajes dirigidos a la comprensión del entorno.
- Se tiene conciencia, pues requieren actividad consciente del sujeto.
- Se obtienen aprendizajes generalmente más duraderos.
- Los aprendizajes son significativos, son fruto de la reflexión.

Aprendizaje significativo

Una vez definido el contexto de desarrollo, mismo que esta basado en el aprendizaje por competencias, el cual a su vez, se apoya en el pilar de la teoría constructivista, es importante definir que tipo de aprendizaje se desea desarrollar. En concordancia con el modelo educativo actual, el aprendizaje significativo (David Ausubel) es el proceso cognitivo que debe desarrollarse en el alumnado a través de los mecanismo adecuados. En general este concepto se define como el conocimiento que apropia por si mismo el estudiante, y que a su vez lo ubica en un espacio de la memoria durable, éste aprendizaje apropiado puede surgir a partir de un principio básico o como producto de información, comportamientos, actitudes o habilidades propias o del medio.

Objetivos de la Investigación

El desarrollo de esta investigación propone el uso de elementos tecnológicos, teóricos y formativos que se precisan en los siguientes objetivos de manera particular.

Educativos

- Documentación y seguimiento de una practica tecnológica encaminada al desarrollo de prototipos electrónicos que permitan acrecentar el acervo formativo y científico del alumnado.
- Establecer una línea de investigación pedagógica que permita elaborar una práctica de tipo transversal en áreas tecnológicas y teóricas de materias afines.
- Tangibilidad de lo abstracto a lo pragmático a través del diseño de nuevas propuesta surgidas de la forma básica de esta práctica.
- Este trabajo va dirigido a estudiantes del Nivel Medio Superior que corresponde a las áreas tecnológicas, pero la proyección considera a las demás ofertas educativas.

Prácticos

- Elaborar un prototipo electrónico básico, que permite visualizar de manera gráfica datos obtenidos a partir de una interface entre un sensor, la placa Arduino, y Processing como intérprete de la información.
- Documentar el desarrollo mecánico, electrónico y digital para su uso en una práctica sustentable de laboratorio.
- Disponibilidad económica de los dispositivos electrónicos y de software.

Competencias Educativas que son aplicables en este Proyecto

Las siguientes competencias pertenecen al SNB que permite establecer un Marco Común Curricular entre los subsistemas y entidades que componen al sector educativo a nivel nacional conforme a los acuerdos SEP.

Competencias Genéricas

Se expresa y comunica

1. Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiados.

Atributos:

1. Expresa ideas y conceptos mediante representaciones lingüísticas, matemáticas o gráficas.
5. Maneja las tecnologías de la información y la comunicación para obtener información y expresar ideas.

Piensa crítica y reflexivamente

2. Desarrolla innovaciones y propone soluciones a problemas a partir de métodos establecidos.

Atributos:

1. Sigue instrucciones y procedimientos de manera reflexiva, comprendiendo como cada uno de sus pasos contribuye al alcance de un objetivo.

Competencias disciplinares

Ciencias Experimentales

4. Obtiene, registra y sistematiza la información para responder a preguntas de carácter científico, consultando fuentes relevantes y realizando experimentos pertinentes.
5. Contrasta los resultados obtenidos en una investigación o experimento con hipótesis previas y comunica sus conclusiones.
9. Diseña modelos o prototipos para resolver problemas, satisfacer necesidades o demostrar principios científicos.

Desarrollo

Práctica de Laboratorio

Se exponen los detalles técnicos de todos los elementos que integran el desarrollo de la propuesta, esto permite conocer tanto conceptos especializados como alcances y limitaciones propias en la implementación de la propuesta de manera práctica en el laboratorio de trabajo.

Componentes de hardware que integran la propuesta (Práctica)

Materiales para realizar el prototipo propuesto

1 Arduino UNO R3 (fig .1).

1 Fotorresistor (LDR) (fig. 2) .

Cables para protoboard.

1 Protoboard.



Fig. 1

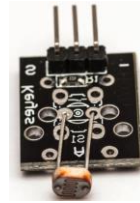


Fig. 2

Arduino, el hardware libre del prototipo electrónico

Esta placa electrónica de código abierto que ha demostrado eficiencia y facilidad de manejo en el desarrollo de prototipos, surge en 2005 como una iniciativa del Instituto de Diseño Interactivo Ivrea en Italia, como una respuesta a la cerrada industria de patentes electrónicos; este hecho consolidó la creación de una plataforma de hardware abierto (open hardware) que actualmente ha sido apoyada por la comunidad científica, y que ha permitido el desarrollo de prototipos creados por jóvenes emprendedores que han consolidado diversas “Start up’s” de tipo electrónico; su facilidad y el soporte dado, hacen de la Placa Electrónico Arduino un excelente aliado en la formación educativa del alumnado.

En cuanto a la programación de la Placa Arduino, el lenguaje de soporte es Wiring, un lenguaje que tiene estructuras semejantes a Java, pero a la vez, simples y transparentes, que han permitido una rápida asimilación por parte de quienes hace uso de ellos; en el caso de esta práctica a desarrollar, dicha ventaja permite una importante ventaja, ya que el alumnado que tenga lecciones básicas de programación en lenguajes tales como Pascal, C, o el propio Java, será capaz de asimilar rápidamente la sintaxis y filosofía en la elaboración de programas para Arduino; lo que no significa que alguien que desconozca de la materia lo delimite, dado que su simplicidad estructural brinda la oportunidad de empezar a programar desde cero. Por

último, Arduino no es un desconocido, ni un proyecto solitario de aficionados a la electrónica, ya que actualmente cuenta con el apoyo de una comunidad de 120 mil usuarios.

Processing, el intérprete gráfico

Es un lenguaje de programación de código abierto basado en Java, Implementado en su momento por Ben Fry y Casey Reas desarrolladores provenientes del MIT Media Lab. El entorno de trabajo de Processing es semejante al IDE de Arduino (Wiring), de hecho este último es basado en gran parte de Processing. Gracias a esta familiaridad se da la oportunidad de usar a Processing con Arduino para desarrollar un entorno de visualización más personalizado de los datos recolectados en el puerto serial a partir de la interface entre sensores y la placa Arduino.

Para programar en Processing será necesario conocer un poco de Java, ya que esta sustentando en este lenguaje y muchas funciones tiene una gran semejanza, por otro lado, como se menciona previamente, la especialidad de Processing es crear entornos visuales gráficos, pero también permite comunicarse con interfaces electrónicas apoyadas en Arduino (Wiring) .

Características de Processing

Tipos de datos:

- byte, short, int, long, float, double, char, boolean, array strings.

Constantes:

- final tipo variable=valor;

Operadores: Aritméticos, Relacionales, Condicionales, Operaciones con bits, asignación.

Ciclos:

while:

- while (expr) { instrucciones }
- do { instrucciones } while (expr)

for

- for (begin; end; inc) {instrucciones }

Condicionales:

if/else

- `if (expr) { instrucciones }`
- `if (expr) { instrucciones } else { instrucciones }`
- switch
 - `switch (var) { case val: instrucciones default: }`

Salto:

- Break, continue, return

Clases, Funciones, Métodos, Propiedades, Atributos:

- Class, length(), entre otros.

La estructura en general de este lenguaje es parecida a la de Arduino y se describe de la siguiente manera.

Instrucción	Descripción	Forma
setup()	<p>Esta instrucción permite la inicialización de los recursos, es decir, en este espacio se podrán definir diversas necesidades del programador: variables, funciones, entre otras; mismas que al finalizar su acción secuencial, emitirán un llamado a la función draw().</p> <p>Es importante mencionar que esta función, sólo se ejecuta una sola vez, y es considerado como el punto de inicio, si alguna vez se ha programado en lenguaje C, setup() podría compararse con main().</p>	<pre>void setup() { // Inicialización // Tamaño lienzo: size() //Fondo color: background }</pre>
draw()	<p>En esta instrucción se agrupan secuencias que se realizarán de manera cíclica hasta terminar el programa. Este es el espacio para todas las instrucciones de dibujo, condicionales o de bucle, así como las comunicaciones.</p>	<pre>void draw() { // Instrucciones que se repiten cíclicamente. }</pre>
size()	<p>Define el espacio de trabajo del lienzo del gráfico. Siendo los parámetros utilizados por esta función, el número de pixeles correspondientes al ancho y alto del la ventana.</p>	<pre>size(200,200)</pre>

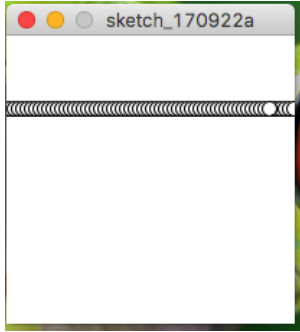
Ejemplo: La siguiente estructura traza de manera constante un círculo generado a partir de una elipse (fig. 3). Un punto importante a mencionar que el hecho de considerar el procesamiento cíclico de la función draw(), por ello se utiliza una estructura un tanto diferente al hecho de solo incrementar.

El incremento de la abscisa esta basado en una división entera (fig. 4), dado que:
Si a la sentencia:

```

sketch_170922a | Processing
1 int xp,h;
2 void setup()
3 {
4   xp = 7;
5   h=200;
6   size(200,200);
7   background(255);
8 }void draw()
9 {
10  xp = (xp+3)%h;
11  ellipse(xp,50,10,10);
12 }
13

```

<ul style="list-style-type: none">▪ $xp = (xp+3)\%h$; <p>se le cambiara por algo tan simple como:</p> <ul style="list-style-type: none">▪ $xp = xp+3$; <p>El hecho de la repetición continua implica un aumento de las coordenadas abscisas, y como consecuencia se rebasa el lienzo.</p>	<p>fig. 3</p>  <p>fig. 4</p>
---	--

Las coordenadas en Processing

El sistema coordenado de pixeles, esta organizado en una retícula rectangular, donde esta estructura permite a Processing representar elementos visuales, los cuales se orientan a través del eje X (abscisa) y el eje Y (ordenadas), ambas en forma consecutiva incremental con origen (0,0) ubicado en la esquina superior izquierda, siendo este espacio definido por la función size().

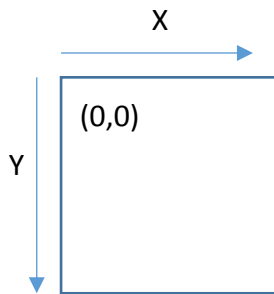


Fig 5. Funciones básicas de dibujo

Instrucción	Descripción
background()	Esta instrucción establece el color de fondo utilizado por la ventana (lienzo) background (rgb) <ul style="list-style-type: none"> rgb: Valor entero de los componentes background (gray) <ul style="list-style-type: none"> gray: Valor entre blanco y negro
fill()	Establece el color de relleno de las formas subsecuentes a su invocación. fill (rgb) <ul style="list-style-type: none"> rgb: Valor entero de los componentes fill (gray) gray: Valor entre blanco y negro
stroke()	Configura el color utilizado para el contorno de las formas. Establece el color de relleno de las formas subsecuentes a su invocación. stroke (rgb) <ul style="list-style-type: none"> rgb: Valor entero de los componentes stroke (gray) gray: Valor entre blanco y negro

Modelo de Color

El modelo de color utilizado frecuentemente esta basado en la mezcla de los colores rojo, verde y azul, denominado Modelo RGB(Red, Green, Blue) el cual no es ajeno a Processing y por ende se puede hacer uso de el para generar propuesta de color en cada elemento gráfico que se considere generar.

En las funciones se puede ingresar de manera paramétrica con los tres valores separados por coma:

- fill(231,123,23);

Además, si necesitara hacer uso de alguna paleta con la codificación de colores, se puede apoyar

al acceder a (fig. 6): Herramientas → Selector de Color (Tools → Color Selector).

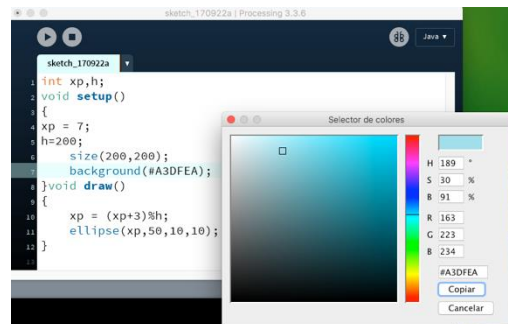


Fig. 6

Interface Arduino y fotoresistor

Fotoreistor (LDR)

Una fotocelda (fotoreistor) o LDR (light-dependent resistor) es un dispositivo cuya resistencia varía en función de la luz recibida de un medio. Esta variación es usada a través de las entradas analógicas de un dispositivo para poder medir una estimación del nivel luminosidad recibida. Un fotoreistor está formado por un semiconductor comúnmente fabricado en sulfuro de cadmio CdS, que al incidir la luz directamente sobre él, los fotones son absorbidos provocando que electrones pasen a la banda de conducción y por ende disminuya la resistencia del componente a medida que aumenta la luz sobre él (fig. 7). Sus valores característicos oscilan entre un 1 μohm en total oscuridad y 50 o 100 Ohm en luz brillante.

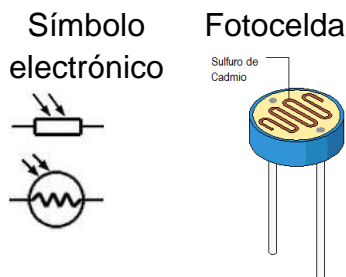


Fig. 7

Material de Fabricación

Es importante conocer el tipo de material fotosensible del que dispone un fotoreistor, ya que su componente (sulfuro de talio, sulfuro de cadmio, sulfuro de plomo o seleniuro de cadmio) determina su uso en la aplicación que se desea desarrollar, esto porque cada material reduce sus resistencia en determinadas bandas de longitud de onda, por ejemplo, una fotocelda construida con sulfuro de plomo sólo es sensible a emisiones infrarrojas, mientras que el sulfuro de cadmio tiene sensibilidad general a todas las radiaciones del espectro visible.

Código Arduino

En el caso de Arduino, la programación para recibir datos de los sensores se basa en “escuchar la entrada de datos” de la entrada Analoga A0, la cual es constantemente censada en un intervalo de 50 ms (milisegundo) para integrar los datos emitidos del fotoresistor al puerto COM activo, mismo que es revisado por Processing para integrar la información generadora del grafico. Por otro lado, es importante mencionar que las señales emitidas por el fotoresistor tienen una salida analógica que debe ser evaluada en un rango específico, que en este caso genera valores entre 0 y 512.

Codificación en el cuerpo cíclico

Función loop()	<pre>void loop() { Luz_Sensor = analogRead(PinSensor); //Lectura desde el sensor (Entrada Analógica) Luz_Sensor=Luz_Sensor/2; Serial.println(); Serial.print(Luz_Sensor); //Enviar al Puerto SERIAL de la PC delay(50); //Pausa entre envios }</pre>
-------------------	--

Codificación en el arranque

Función setup() Se establecen las condiciones de arranque, sólo se ejecuta una vez.	<pre>void setup() { Serial.begin(9600); //Puerto de comunicación -vel 9600 bps }</pre>
--	--

Interface entre Arduino (wiring) y Processing

Como se menciona anteriormente, para generar una interpretación gráfica en pantalla a través de Processing, será necesario recolectar la información recibida en el puerto COM activo, el cual recibe datos provenientes del fotoresistor que a su vez son organizados por Wiring con la programación la placa Arduino, el esquema se presenta de la siguiente manera(fig. 8).

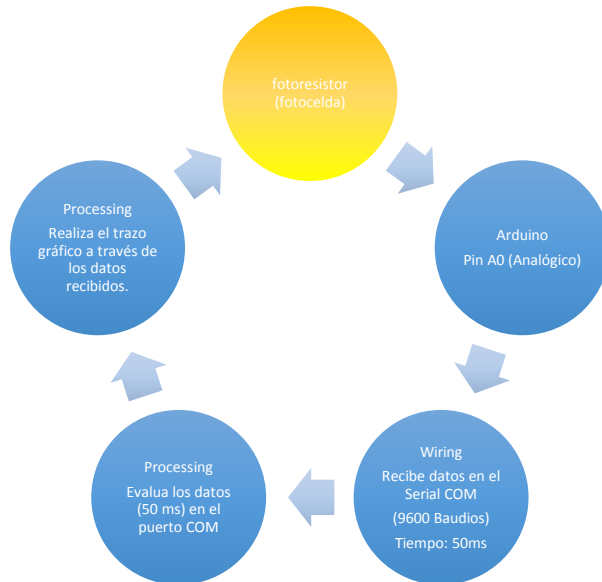


Fig. 8 Ciclo de interfaz Arduino-Processing

Código Processing

La programación es muy simple, una vez obtenidos los datos del puerto COM activo, se establece un área de diseño donde se trazarán líneas rectas que denotarán por su altura con respecto a un origen la intensidad de luz que ingresa al fotodiodo vía puerto COM, además de texto ilustrativos; hay un punto importante que sobresale, tanto Processing como Wiring son programas cíclicos, por lo tanto se debe planificar condiciones dentro del cuerpo de la función repetitiva para realizar la gráfica adecuadamente.

En este caso, la condición se encuentra en la función draw() a través de una estructura condicional simple (if-else), que evalúa lo siguiente:

- Disponibilidad del puerto.
- Conversión de los datos (Cadena a valor)
- Avance de las coordenadas de trazo

Codificación en el cuerpo cíclico	
Función draw()	<pre> void draw() { while (myPort.available() > 0) { myStr = myPort.readStringUntil(lf); if (myStr != null) { num=int(myStr); // Conversión string a entero println(num); fill(#0000ff); if (xp+step>=width-50){ xp=20; step=0; background(255,255,255); } else{ step=step+3; } stroke(27, 24, 25); ellipse(xp+step,yp-num,1,1); line(xp+step,yp-num,xp+step,yp); ... } </pre>

Codificación en el arranque	
Función setup() Se establecen las condiciones de arranque, sólo se ejecuta una vez.	<pre> void setup() { background(255,255,255); size(600,600); myPort = new Serial(this, Serial.list()[2], 9600); //"COM5" myPort.clear(); } </pre>

Observaciones Importantes

Durante el desarrollo práctico del prototipo se dieron los siguientes casos, situación que es importante mencionar, debido a que existe la posibilidad de incurrir en ellos.

- Se obtiene el error: `ArrayIndexOutOfBoundsException:#`

En caso de no identificar adecuadamente el Puerto COM que permite comunicarse con la Placa Arduino, puede definir la cadena que corresponde, por ejemplo escriba “COM5” como sustituto de la instrucción `Serial.list(0)[0]`.

- Se obtiene el error: `Error opening serial port COM#: Port Busy`

La razón de este mensaje se debe a que no puede estar abierto el Serial Monitor mientras se ejecuta una rutina de Processing que necesita acceder al mismo Puerto.

Diagrama de la práctica (Prototipo)

En la implementación se hace uso de un fotoresistor encapsulado como IC, al cual se le proporciona la energía eléctrica necesaria y de ella se suministran datos que ingresan a A0 (fig 9). Pero en atención a la ausencia de un fotoresistor IC en el mercado, se propone una “circuito alternativo” que permite realizar la misma actividad.

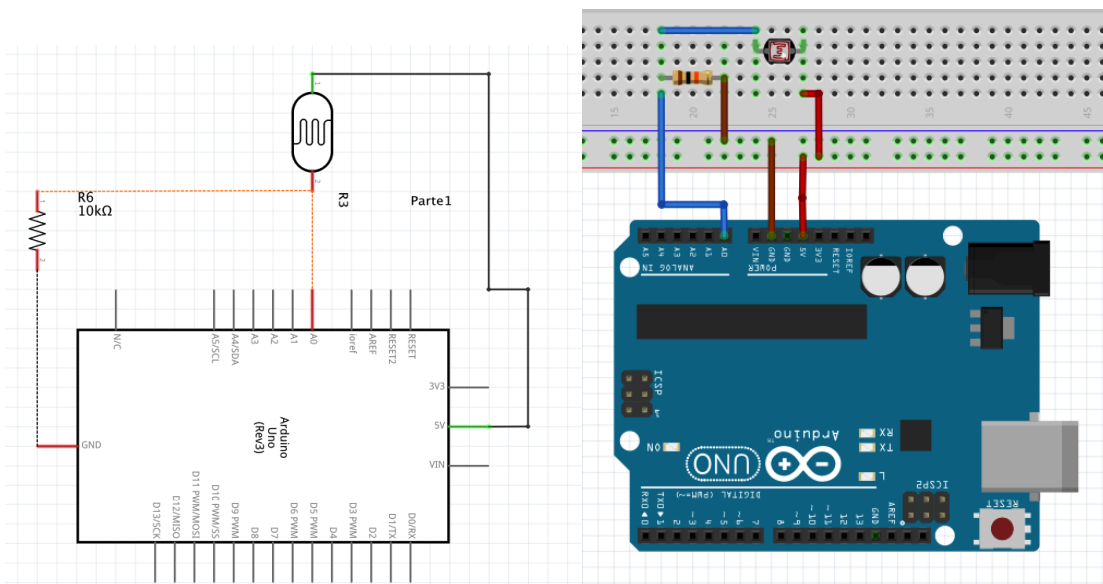


Fig. 9

Resultados

El prototipo elaborado de manera física se presenta a continuación, en el se aprecia un fotoresistor como componente integrado, que de igual forma funciona con un armado basado en resistencias. El sensor recibe la intensidad de luminiscencia y genera un resultado enviado a la entrada analógica A0 de la Placa Arduino. Los datos recibidos son utilizados por Processing y generan un gráfico de áreas que permite apreciar la intensidad de luz recibida a través de los datos recolectados en el puerto COM activo.

El prototipo básico se presenta en la siguiente imagen y muestra la conexión física de los componentes Arduino + fotoresistor (fig. 10 y 11).

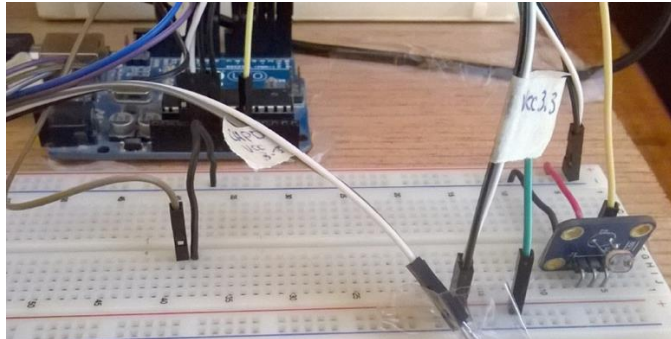


Fig. 10

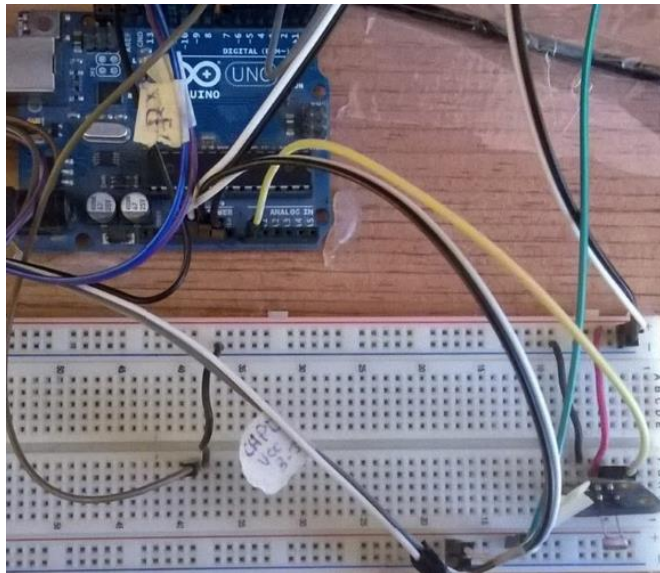


Fig. 11

A continuación se expone una captura de pantalla, que presenta la forma visual de los datos recabados del puerto COM y su interpretación con Processing a través de un gráfico de “barras”.

En este gráfico se aprecia una luminiscencia ambiental que genera un trazo continuo de las barras sobre la escala ilustrada (fig 12).

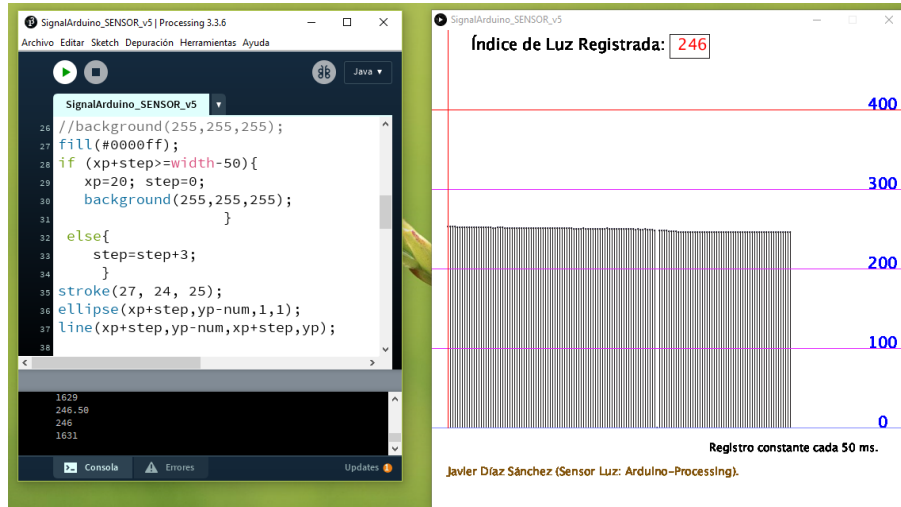


Fig 12.

Cuando existe una variación en la presencia de luz sobre el fotoresistor, el ingreso de datos sobre la entrada analógica del pin A0 de la Placa Arduino, se recolectado para su trazo como grafico a través de Processing, en este ejemplo se muestra el comportamiento ante esta variación luminaria. Se pueden apreciar altibajos y constancia en los trazos derivados del censado de datos del puerto COM activo (fig. 13).

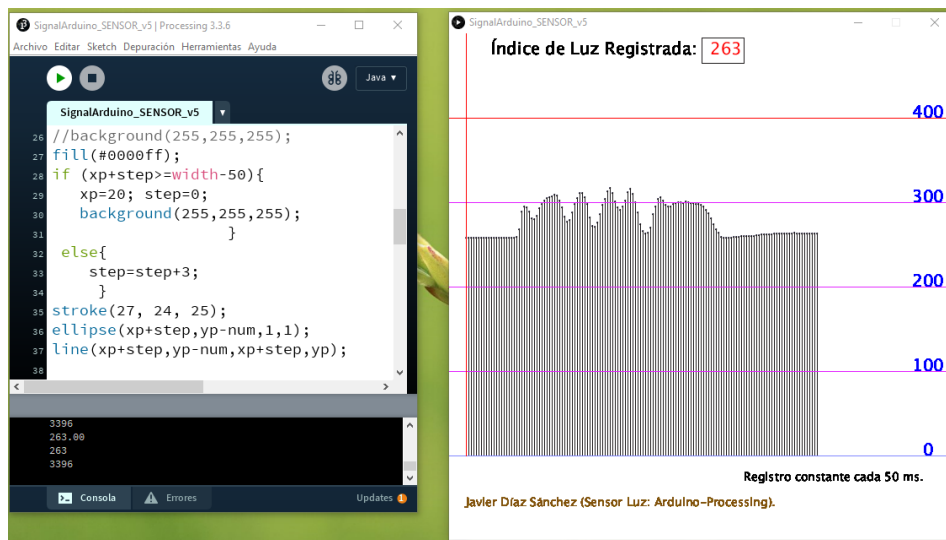


Fig. 13

Atributo	Competencias		Competencias Disciplinarias		
	Genérica 4	Genérica 5	Experimental 4	Experimental 5	Experimental 9
1	<p>Interpreta conceptos de electrónica básica</p> <p>Utiliza código y manuales en el manejo de circuitos y componentes (resistencias, diodos, leds).</p>	<p>A partir de los conocimientos adquiridos en el manejo de los componentes y circuitos, elabora soluciones que responden a necesidades de interfaz entre el mundo analógico y el digital.</p> <p>Desarrolla una interface hardware-software y visualiza los datos registrados de manera gráfica.</p>	<p>Consulta manuales de IC, Arduino, Componentes y Programación</p> <p>Para llevar a cabo pruebas y ensamblaje.</p>	<p>Experimenta y Comprueba a través de pruebas de laboratorio en forma responsable.</p>	<p>Elabora el ensamblaje del proyecto, manufactura o propone mejoras al prototipo, ya sea de manera mecánica, digital o informática.</p> <p>Interpreta los resultados obtenidos.</p>
5	<p>Utiliza medios digitales e internet para buscar información sobre hojas de datos aplicados a integrados diversos y componentes.</p>				

Tabla de Competencias desarrolladas por el alumnado en el desarrollo de la propuesta.

A continuación se presenta una tabla que indica que competencias son desarrolladas durante la implementación de la propuesta, en ella se describen cada uno de estos procesos y su correlación con la competencia del Nivel Medio superior bajo la RIEMS y el SNB.

Conclusión

El Nivel Medio Superior a través de la RIEMS, se estable la necesidad de una enseñanza significativa, pragmática y teórica, capaz de resolver problemáticas de la vida diaria, pero también la necesidad de esquematizar propuesta a través de las bases teóricas del conocimiento. Es una invitación a la generación de ciencia con conciencia de causa, una movilidad de saberes con objetivos tangible; con todo lo anterior, esta propuesta en su forma simple, permite una transversalidad de saberes que se enmarca en los propósitos de las competencias propias de la reforma, pero además permiten presentar un prototipo electrónico capaz de medir el medio en una forma de interpretación como lo es la luminiscencia, además permite hacer una análisis de los datos arrojados para una interpretación visual de ese fenómeno llamado luz. Un hecho que no sólo implica desarrollar una práctica, sino conciencia con la práctica surgida de las necesidades del contexto, de hecho, elaborar toda la propuesta implicó el manejo de diversos factores tanto electrónicos como de programación, ya que no sólo se involucró de manera aislada el manejo de la Placa Arduino, sino que fue necesario adquirir conocimientos de otro lenguaje que se especializa en formas graficas como es el caso de Processing, lo que significó un pequeño reto. Finalmente, para llevar acabo cualquier innovación en la educación, es necesario del apoyo de las diversas autoridades que integrar el centro educativo, para convertir su implementación y difusión como una práctica curricular y no sólo como un evento aislado.

Bibliografía

Brian W. Evans (2008). Arduino Programming Notebook: A Beginner's Reference, California, USA:GPL.

Miguel Pareja Aparicio(2015). Iniciación a Arduino Uno. MARCOMBO, S.A. 978-8426721457.

Jonassen, D. H. (1996). Learning with Technology: Using Computers as Cognitive Tools. En D.H Jonassen, Handbook of Research for Educational Communications and Technology (pp. 693 - 719). New York: Macmillan. Recuperado de <http://www.uoc.edu/rusc/5/2/dt/esp/hernandez.pdf>.