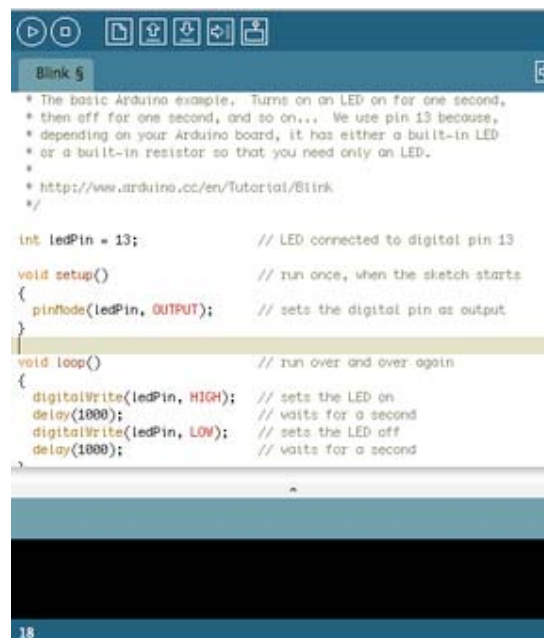


Capítulo 4

Presentación de Datos

4.1 Processing

En cada proceso del programa anterior cada vez que se realiza la medición de una variable esta se manda a imprimir por medio del puerto serial, pero presentar estos datos en un pequeño espacio en una pantalla negra no es tan conveniente.



```
Arduino IDE - Blink 5
* The basic Arduino example. Turns on an LED on for one second,
* then off for one second, and so on... We use pin 13 because,
* depending on your Arduino board, it has either a built-in LED
* or a built-in resistor so that you need only an LED.
*
* http://www.arduino.cc/en/Tutorial/Blink
*/

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Serial Monitor (black background):

18

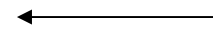


Figura 4. 1: Software Arduino.

Lo mejor es presentar los datos que se van obteniendo por medio de una pantalla donde se puedan visualizar y para esto se busco información [21] en donde usa el programa “Processing” para realizar la representación de los datos [22].

“Processing” es un lenguaje y entorno de programación de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab.” [23].

Por medio de este código se los datos se pudieron mostrar de esta manera:

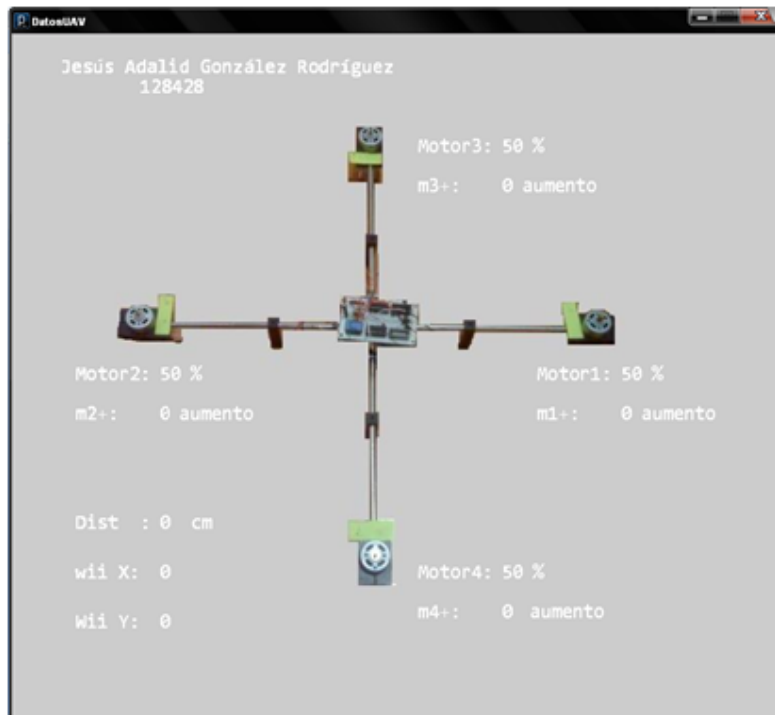


Figura 4. 2: Datos por medio del programa Processing.

Para poder representar las variables por medio del programa hay que cambiar la configuración en la que estas son mandadas al puerto serial, antes se mandaba el valor de alguna variable con un comentario para saber de qué variable se trataba, en este caso para poder identificar cada variable solo basta acompañarla con una letra identificadora, la cual por medio del programa será analizada con su valor en el código ASCII y será reasignada para ser desplegada posteriormente. El siguiente código ilustra cómo se realizó la reasignación de variables:

```
void serialEvent(int serial)           // INICIO DEL METODO
serialEvent
{
  if (serial == a)                     // SI LA ENTRADA DE SERIAL ES
  UN +
  {
    val = Integer.parseInt(buff);      // SE TOMA EL STRING QUE
    ESTA EN BUFF Y SE ENVIA A VAL
    buff = "";                         // SE LIMPIA BUFF
  }

  else if (serial == b)                // SINO, SI LA ENTRADA DE
  SERIAL ES UN -
  {
    val1 = Integer.parseInt(buff);     // SE TOMA EL STRING QUE
    ESTA EN BUFF Y SE ENVIA A VAL3
    buff = "";                         // SE LIMPIA BUFF
  }

  else if (serial == c)
  {
    val2 = Integer.parseInt(buff);
    buff = "";
  }

  else if (serial == d)
  {
    val3 = Integer.parseInt(buff);
    buff = "";
  }
}
```

```

else if (serial == e)
{
    val4 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == f)
{
    val5 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == g)
{
    val6 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == h)
{
    val7 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == i)
{
    val8 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == j)
{
    val9 = Integer.parseInt(buff);
    buff = "";
}

else if (serial == k)
{
    val10 = Integer.parseInt(buff);
    buff = "";
}

else
{
    buff += char(serial);
}
//DE LA ENTRADA SERIAL EN EL BUFF
}

//no es un carácter
// SE ALMACENA EL CHARACTER

```

En esta parte de código se muestra como se realizó la asignación de las variables obtenidas por medio del puerto serial y asignadas a un parámetro de medición, la asignación de todas las variables usadas es la siguiente:

Tabla 1: Re asignación de variables para el Processing

Variables	Asignación
a	Motor 1
b	Motor 2
c	Motor 3
d	Motor 4
e	Distancia en cm
f	Inclinación en X
g	Inclinación en Y
h	Incremento M1
i	Incremento M2
j	Incremento M3
k	Incremento M4

El código completo de asignación de variables en el Processing se muestra en el Apéndice F¹.

¹ Apéndice F ubicado en la página 102.