

Aplicación del micrófono multi-arreglo de Kinect y android para control de acciones.

*Estrada C. **Diaz A., *** Quitero V.

¹Resumen— *En este documento se analiza la programación y control de sistemas mediante herramientas; i) App Inventor para controlar acciones a un robot. Actualmente la programación de aplicaciones propias utilizando Android, es un campo abierto para el cómputo móvil, nosotros utilizamos el software App inventor del Massachusetts Institute of Technology para Google [6] el cual facilita la implementación de aplicaciones móviles. También se implementó el control, mediante ii) Cámara Kinect (Xbox) [2] la cual manipula acciones mediante voz en Tiempo Real.*

La cámara Kinect actúa como un sensor auditivo. Todo esto no sería posible sin la utilización del micro controlador iii) Arduino, el cual es la interfaz para lectura de datos, procesarlos y dar acciones a un conjunto de servomecanismos. Arduino es flexible, fácil de utilizar y es una plataforma abierta para la creación de prototipos y objetos interactivos.

El nuevo periférico de la Xbox 360, Kinect, cuenta con un micrófono multirray que es capaz de separar las voces que hay justo delante del resto de sonidos del entorno para chatear y utilizar comandos de voz.

Palabras claves: Arduino, Kinect, Programación en tiempo real, Control, Microcontrolador, App Inventor, Android, micrófono multi-arreglo.

I. INTRODUCCIÓN

Este proyecto analiza el diseño de un sistema para el manejo de un brazo robótico, sin la utilización de controles manuales, evitando el contacto directo.

Se optó por manejar un brazo robótico ya que contiene componentes mecánicos y electrónicos, para efectuar tareas utilizadas en la industria y en la educación [8].

*Carlos de Jesús Estrada Rentería del Estudiante Instituto Tecnológico de Mexicali Av. Tecnológico S/N col Elías Calles B.C
Estradacdj@gmail.com

**Dr. Arnoldo Díaz Ramírez Profesor- investigador del Instituto Tecnológico de Mexicali Av. Tecnológico S/N col Elías Calles B.C
Adiaz@itmexicali.edu.mx

***Verónica Quintero Rosas Profesor del Instituto Tecnológico de Mexicali Av. Tecnológico S/N col Elías Calles B.C
veroquintero@yandex.ru

Para hacer este diseño se estudió la cinemática del robot el cual estudia los movimientos respecto a un sistema de referencia. La cinemática describe analíticamente el movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares.

El control auditivo al ser a distancia nos da una protección más del usuario al material a manejar, por medio de un conjunto de ordenes al sensor auditivo que utilizamos, Kinect por ser un micrófono multi arreglo, lo cual sirve de receptor y detecta la voz del usuario entre ruido y otras voces, lo que hace de este sensor as eficaz que un micrófono convencional, además su rango de alcance y con ayuda de librerías de uso libre podemos desarrollar un control libre del contacto.

Se tendrá control de un brazo robótico, por medio de órdenes por comando de voz, que serán procesados y enviados a un controlador en tiempo real y ejecutará la orden, en este caso se manejaran tres tipos de ordenas, desde reposo, saludar y bailar.

II. APP INVENTOR (MIT).

La utilización común de android es mediante la programación en eclipse, sin embargo se decidió utilizar el app inventor debido a su facilidad, rango de operación rápida y robusta.

App Inventor [6] es una aplicación de Google Labs para crear aplicaciones de software para el sistema operativo Android de forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir dando una serie de bloques para crear la aplicación. Estos bloques son muy parecidos a la programación en bloques de matlab e inclusive de Lab View. Es muy amigable ya que algunas universidades lo comparan con Visual Basic, en la forma de programar mediante botones.

Las librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License).

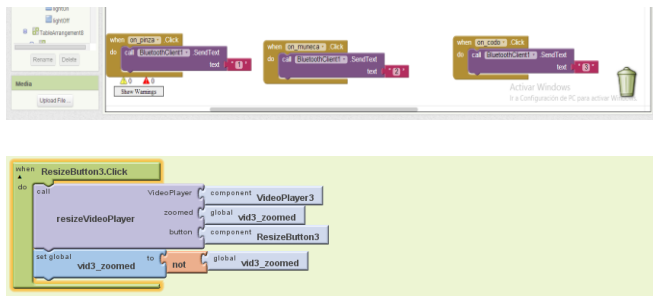


Fig. 1 Visor

III. ARDUINO.

Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino solo pueden ejecutarse sin necesidad de conectar a un ordenador.

El software puede ser descargado de forma gratuita. Arduino es una plataforma de hardware libre basada en una placa con un microcontrolador diseñada para facilitar el uso de la electrónica en general consta de puertos de entrada y salida. Arduino solo es la interfaz de comunicación quien dará en las salidas las acciones a los servomecanismos.

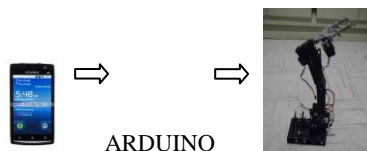


Figura 2. Diagrama de bloques interacción Celular-robot

IV. KINECT.

Kinect, es un controlador de juego libre desarrollado por Microsoft para la videoconsola Xbox 360, Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador, mediante una interfaz que reconoce gestos, comandos de voz, y objetos e imágenes. El sensor de Kinect es una barra horizontal de aproximadamente 23 cm. [3]

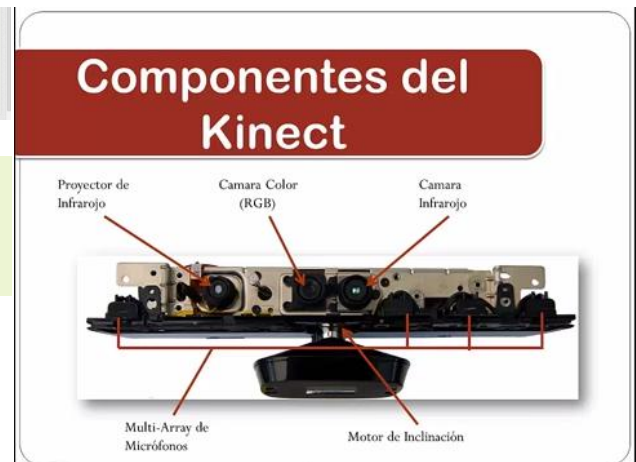


Figura 3 Diagrama componentes de Kinect.

a) Cámara.

El dispositivo cuenta con una cámara RGB, un sensor de profundidad, un micrófono de múltiples matrices y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de

todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz. El micrófono de matrices del sensor de Kinect permite a la Xbox 360 llevar a cabo la localización de la fuente acústica y la supresión del ruido ambiente. [3]

El sensor de profundidad es un proyector de infrarrojos combinado con un sensor monocromo que permite a Kinect ver en 3D. El hardware de Kinect se ha confirmado que se basará en un diseño de referencia y la tecnología 3D-color fabricados por la compañía israelí de desarrollo PrimeSenseLtd. Para poder manipular Kinect se tienen que exportar librerías y entender los software de Processing, OpenNI, y PrimesenseNite; a continuación se explica cada uno de ellos. [3]

b) Micrófono multi-arreglo.

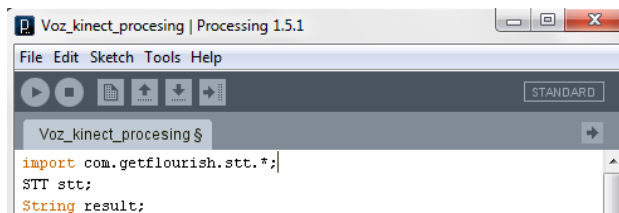
Un micrófono multiarray son muchos micrófonos colocados en fila alrededor de una superficie captando, todos ellos, el sonido que les llega en todas las direcciones. En el caso de Kinect tenemos, 4 micrófonos en línea, tres en el lado izquierdo y otro pegado a la derecha, todos ellos situados en la parte de abajo del

dispositivo. Como es lógico, si situamos micrófonos en distintos lugares, el sonido les llegará a éstos en instantes diferentes, de forma que, a groso modo, se puede calcular de dónde viene la fuente de sonido si se tiene en cuenta este desfase entre las señales que captan los micrófonos y la velocidad del sonido en el aire. Pero no sólo se puede calcular si el sonido viene de un lado o de otro, también se puede determinar aproximadamente su posición. [4]

V. PRORAMACION DE VOZ.

Procesing [7] es un lenguaje de programación, inicialmente creado para servir como un cuaderno de bocetos para enseñar fundamentos de programación de computadoras dentro de un contexto visual. Puede descargarse gratis y de código abierto basado en Java. [2]

Para que Procesing controle la señal auditiva entrante, se necesita la librería STT[5] la cual es el que transcribir y llamar a la entrada la cual puede ser utilizada a conveniencia. La llamada a la librería STT:



Los datos obtenidos por Procesing y STT, generan una respuesta que se envía directamente a Arduino el cual, envía una serie de órdenes a los servomotores.

La interface es vía USB usando así el serial 9600:

```

void setup ()
{println(Serial.list());
  String portName=Serial.list()[0];
  port=new Serial(this,portName,9600)
  
```

El código para Procesing que se encarga de la recepción de datos por micrófonos:

```

stt = new STT(this);
stt.enableDebug();
stt.setLanguage("es");
stt.enableAutoRecord();
// muestra el texto
textFont(createFont("Arial", 24));
result = "Set order";
  
```

Puesto que ya se tiene en la variable “result”, la ordenada, este mismo la traduce en un código de valor,0,1 o 2 la cual es enviada al controlador Arduino.

```

void draw () {
  background(0);
  text(result, mouseX, mouseY);
  if(result.equals("saluda") == true){
    order=1;
  }
  if(result.equals("baila") == true){
    order=2;
  },
  if(result.equals("reposo") == true){
    order=0;
  }
  byte out[]=new byte[1];
  out[0]= byte(order);
  port.write(out[0]);
  
```

Nótese que el valor se escribe en la variable “order” y la misma enviada por el puerto USB. El último paso es Arduino, que se encargara de mandar las señales a los servomotores donde declaramos la librería de serial USB, servomotores, y declaramos las variables:

```
#include <SoftwareSerial.h>
#include <Servo.h>
// declaracion de los servos
Servo torso;
Servo hombro;
Servo codo;
Servo muñeca;
Servo mano;
Servo dedos;
int led=12;
int servoAngulo;
```

Se seleccionaron las salidas para cada uno de los servos y se hace llamar al puerto Serial USB [7]:

```
void setup()
{
  // determina los pines de cada servo
  pinMode (led,OUTPUT);
  torso.attach(2);
  hombro.attach(3);
  codo.attach(4);
  muñeca.attach(5);
  mano.attach(6);
  dedos.attach(7);
  // Tiempo de respuesta de la interface
  Serial.begin(9600);
}
```

Hacemos notar en el siguiente código como con la orden de reposo, la variable orden será “0”, lo que tendremos como resultado un ángulo en los servomotores 90° sin que su posición cambie:

```
void loop () {
  //cumpliendo el serial
  if(Serial.available()){
    digitalWrite(led,HIGH);
    Orden=Serial.read();
    if (Orden==0)
    {
      torso.write(90);
      hombro.write(90);
      codo.write(90);
      muñeca.write(90);
      mano.write(90);
      dedos.write(90);
    }
  }
}
```

Como el mando de reposo esta estático no conlleva mucho algoritmo el mantener los servos inmóviles, pero

si notamos en la siguiente orden, Nótese el cambio y se modifica el código:

```
Orden=Serial.read();
if (Orden==1)
{
  int n=0;
  torso.write(90);
  hombro.write(90);
  codo.write(90);
  mano.write(90);
  dedos.write(90);
  for (n=90; n<180;pos+=10) // De reposo a 180°pasos de 10°
  {
    muñeca.write(n);
    delay (30); //espera 30ms cada 10°para dar tiempo a llegar
    // ala posicion.
  }
  for (n=180;n>=90; pos-=10)// va de 180 a 90 grados
  {
    muñeca.write(n);
    delay (30);
  }
}
```

VI. REFERENCIAS

- [1] <http://www.arduino.cc/es/>
- [2] <http://processing.org/>
- [3] <http://www.xbox.com/es-MX/Kinect>
- [4] <http://www.pensamientoscomputables.com/entrada/Kinect/microfono/multiarray/como-funciona/xbox-360>
- [5] <http://stt.getflourish.com/>
- [6] <http://appinventor.mit.edu/explore/>
- [7] <http://www.processing.org/reference/libraries/serial/>
- [8] Barrientos, Peñin, Balaguer, Aracil “Fundamentos de Robótica”

VII. RECONOCIMIENTOS

Carlos de Jesús Estrada Rentería, agradece a dos personas que le han apoyado en su desarrollo como estudiantes, desarrollador y persona, no desistir y a seguir adelante en lo que desea, M.C Verónica Quintero quien a brindando el apoyo y dado los ánimos para seguir adelante en proyectos e ideas, y Karen Yesenia Noris Mares quien nunca deja que mi cara vea el suelo, y menos tocarlo. Gracias