



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**UNIDAD DE GESTIÓN DE  TECNOLOGÍAS**

**CARRERA DE ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN & AVIÓNICA**

Autor: Zapata Toapanta Edwin Israel


**TEMA: IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL PARA UN BRAZO ROBÓTICO EN BASE A UNA INTERFAZ ANDROID-ARDUINO PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

# OBJETIVO GENERAL

- Implementar un prototipo de control para un brazo robótico en base a una interfaz Android-Arduino para el laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías-Universidad de las Fuerzas Armadas-ESPE.

# OBJETIVOS ESPECÍFICOS

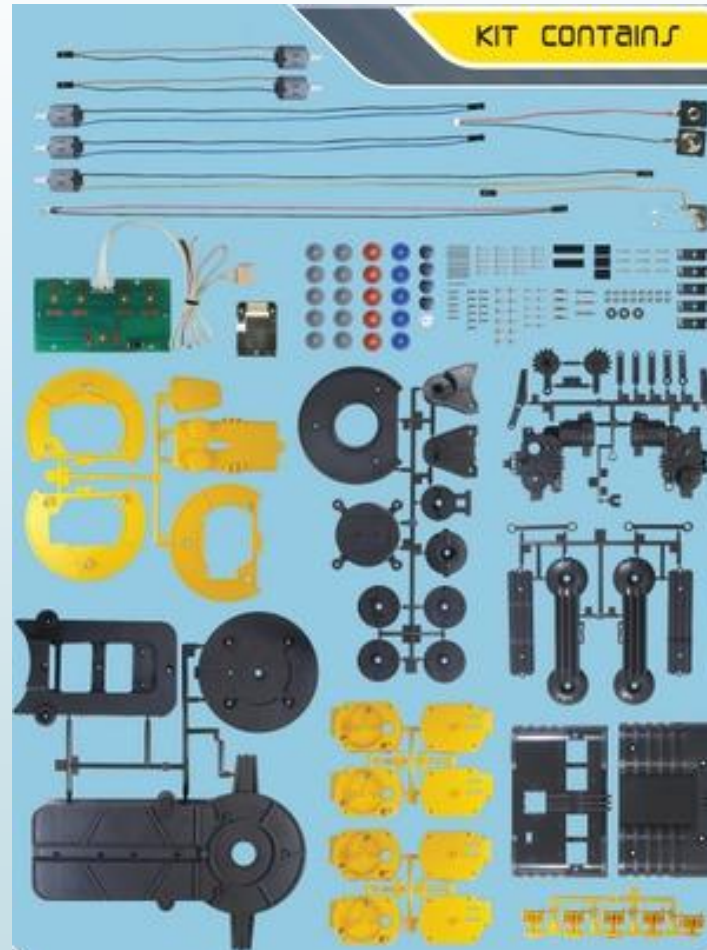
- Realizar un análisis de la tecnología Arduino y software de configuración que pueden ser utilizados en la implementación de robots.
- Establecer una comunicación bluetooth con la ayuda de la tarjeta Arduino MEGA y el módulo bluetooth HC-06 entre el teléfono móvil y los actuadores de los robots.

- 
- Analizar los principales comandos de programación de App Inventor para diseñar una aplicación que al ejecutarla desde un celular Android sea capaz de controlar a los brazos robóticos.
  - Desarrollar un prototipo de proceso industrial que permita controlar un brazo robótico controlado desde una aplicación instalada en un smartphone.

# ENSAMBLAJE DE LOS BRAZOS ROBÓTICOS



# PARTES PIEZAS Y COMPONENTES

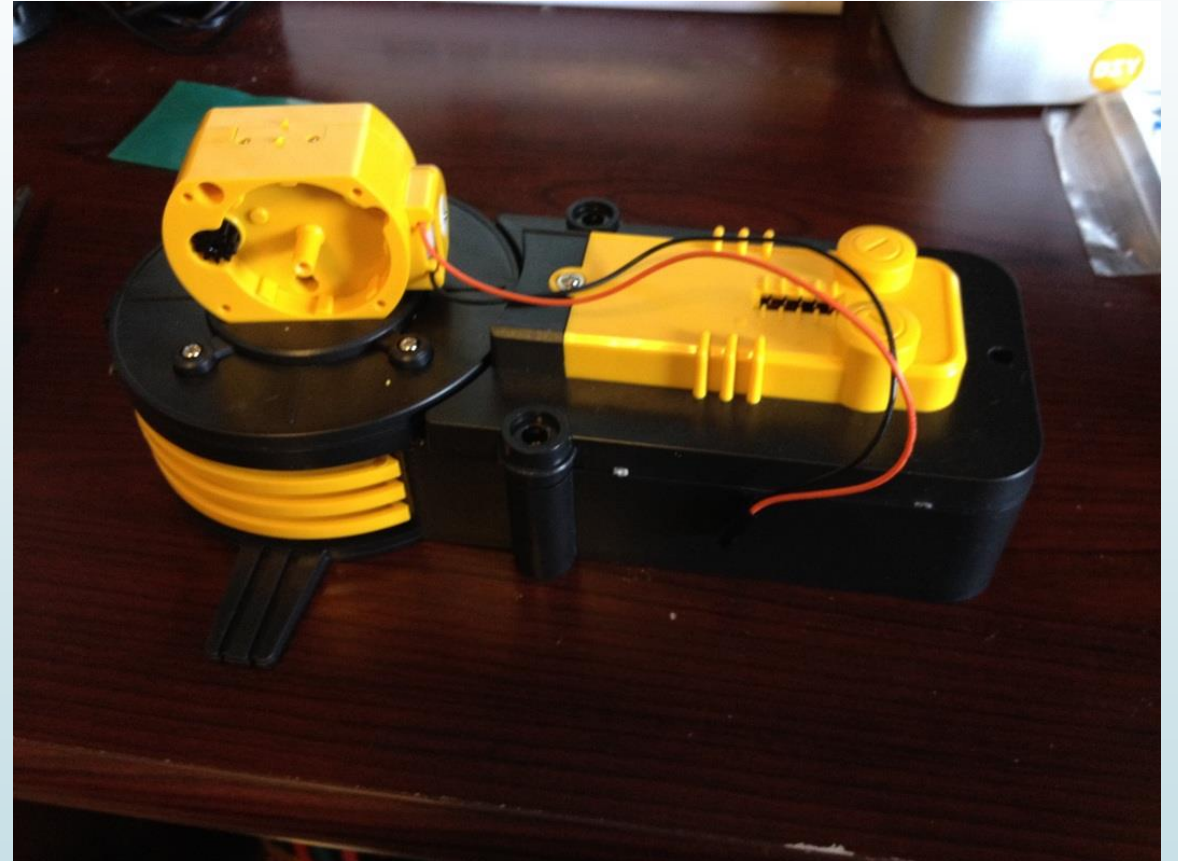




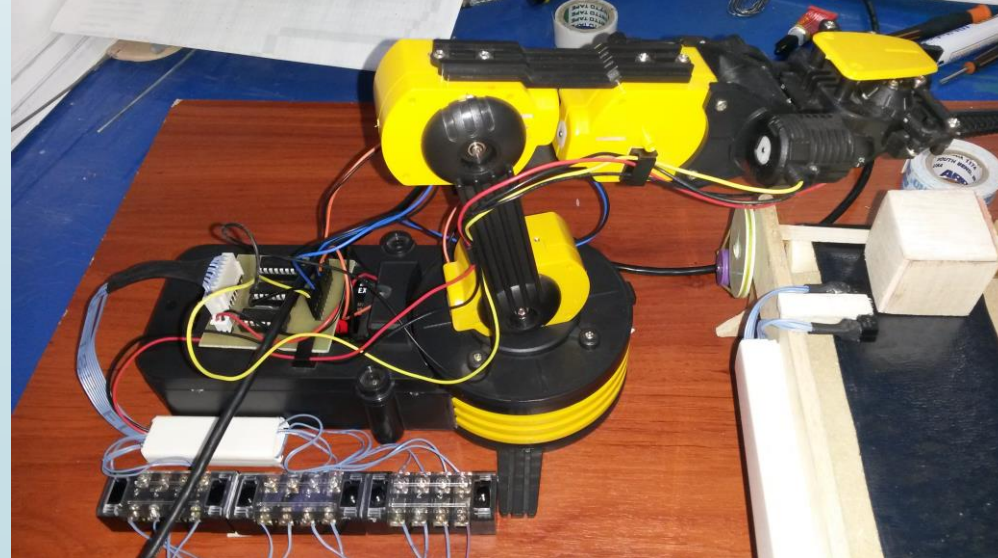
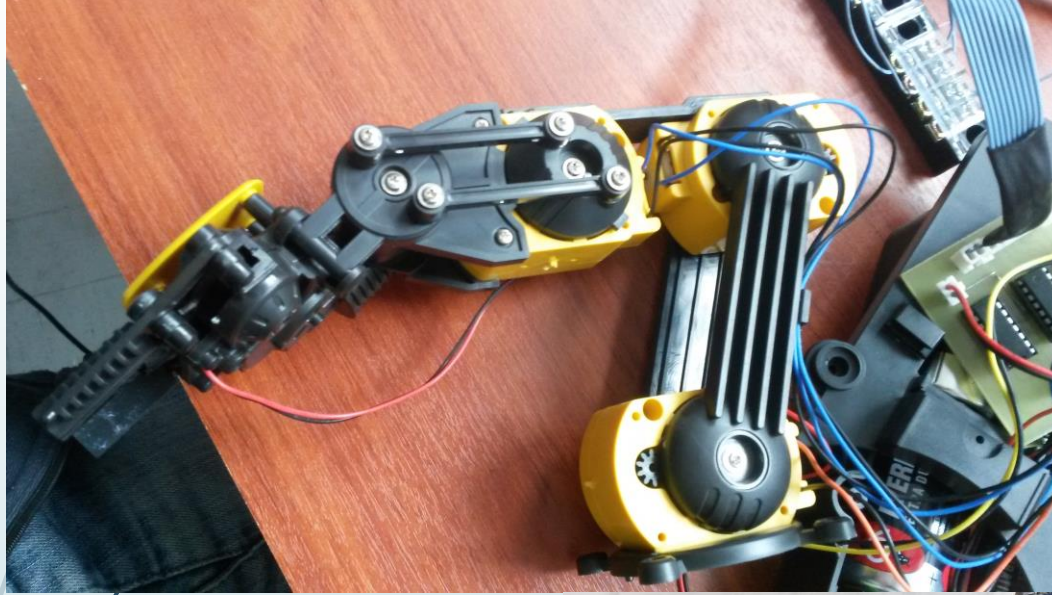




# CAJAS MOTOREDUCTORAS







# PROGRAMACIÓN ARDUINO



AN OPEN PROJECT WRITTEN, DEBUGGED AND SUPPORTED  
BY MASSIMO BANZI, DAVID CUARTIELLES, TOM IOOE,  
GIANLUCA MARTINO AND DAVID MELLIS

BASED ON PROCESSING BY CASEY REAS AND BEN FRY



# ASIGNACIÓN DE PUERTOS

```
const int HA = 22; //HOMBRO
const int HB = 23; //HOMBRO
const int BA = 24; //ANTEBRAZO
const int BB = 25; //ANTEBRAZO
const int AA = 26; //BRAZO
const int AB = 27; //BRAZO
const int MA = 28; //MUÑECA ARRIBA
const int MB = 29; //MUÑECA
const int PA = 30; //PINZA
const int PC = 31; //PINZA
//BRAZO 2
const int HA_B = 32; //HOMBRO
const int HB_B = 33; //HOMBRO
const int BA_B = 34; //ANTEBRAZO
const int BB_B = 35; //ANTEBRAZO
const int AA_B = 36; //BRAZO
const int AB_B = 37; //BRAZO
const int MA_B = 38; //MUÑECA ARRIBA
const int MB_B = 39; //MUÑECA
const int PA_B = 40; //PINZA
const int PC_B = 41; //PINZA

int SENSOR1 = A0;
int SENSOR2 = A1;
```

```
int SENSOR1 = A0;
int SENSOR2 = A1;

int I1 = 0;
int I2 = 0;

int LEDA = 49;
int LEDB = 48;

const int MOTOR = 52;
const int MOTOR1 = 53;
```

## INICIO DE LA COMUNICACIÓN

```
void setup() {
  Serial.begin(9600);
```

# CONFIGURACIÓN DE PUERTOS

```
pinMode(HA, OUTPUT);
pinMode(HB, OUTPUT);
pinMode(BA, OUTPUT);
pinMode(BB, OUTPUT);
pinMode(AA, OUTPUT);
pinMode(AB, OUTPUT);
pinMode(MA, OUTPUT);
pinMode(MB, OUTPUT);
pinMode(PA, OUTPUT);
pinMode(PC, OUTPUT);
//LED A
pinMode(LED A, OUTPUT);
//BRAZO B
pinMode(HA_B, OUTPUT);
pinMode(HB_B, OUTPUT);
pinMode(BA_B, OUTPUT);
pinMode(BB_B, OUTPUT);
pinMode(AA_B, OUTPUT);
pinMode(AB_B, OUTPUT);
pinMode(MA_B, OUTPUT);
pinMode(MB_B, OUTPUT);
pinMode(PA_B, OUTPUT);
pinMode(PC_B, OUTPUT);
```

```
//LED B
pinMode(LED B, OUTPUT);
//SENSORES
pinMode(SENSOR1, INPUT);
pinMode(SENSOR2, INPUT);

// BANDA TRANSPORTADORA
pinMode(MOTOR, OUTPUT);
pinMode(MOTOR1, OUTPUT);
```



# ENVÍO DEL CÓDIGO

```
if (readString == "A") {
    digitalWrite(HA, HIGH);
    digitalWrite(HB, LOW);
    delay(100);
    digitalWrite(HA, LOW);
}
else if (readString == "B") {
    digitalWrite(HA, LOW);
    digitalWrite(HB, HIGH);
    delay(100);
    digitalWrite(HB, LOW);
}
}

// CONTROL BRAZO PINES 24 Y 25
if (readString == "C") {
    digitalWrite(BA, HIGH);
    digitalWrite(BB, LOW);
    delay(100);
    digitalWrite(BA, LOW);
}
else if (readString == "D") {
    digitalWrite(BA, LOW);
    digitalWrite(BB, HIGH);
    delay(100);
    digitalWrite(BB, LOW);
}
```

```
if (readString == "K") {
    digitalWrite(HA_B, HIGH);
    digitalWrite(HB_B, LOW);
    delay(100);
    digitalWrite(HA_B, LOW);
}
else if (readString == "L") {
    Serial.println("Izquierda");
    digitalWrite(HA_B, LOW);
    digitalWrite(HB_B, HIGH);
    delay(100);
    digitalWrite(HB_B, LOW);
}
}

// CONTROL BRAZO PINES 34 Y 35
if (readString == "M") {
    digitalWrite(BA_B, HIGH);
    digitalWrite(BB_B, LOW);
    delay(100);
    digitalWrite(BA_B, LOW);
}
else if (readString == "N") {
    Serial.println("ABAJO");
    digitalWrite(BA_B, LOW);
    digitalWrite(BB_B, HIGH);
    delay(100);
}
```

```
if (readString == "U") {
    digitalWrite(LED_A, HIGH);
    digitalWrite(BA, HIGH);
    delay(1500);
    digitalWrite(BA, LOW);
    digitalWrite(AB, HIGH);
    delay(2500);
    digitalWrite(AB, LOW);
    digitalWrite(MB, HIGH);
    delay(1250);
    digitalWrite(MB, LOW);
    digitalWrite(PC, HIGH);
    delay(1650);

    //retroceso
    digitalWrite(PC, LOW);
    digitalWrite(MA, HIGH);
    delay(1500);
    digitalWrite(MA, LOW);
    digitalWrite(AA, HIGH);
    delay(2750);
    digitalWrite(AA, LOW);
    digitalWrite(BB, HIGH);
    delay(1500);
    digitalWrite(BB, LOW);
    digitalWrite(MB, HIGH);
    delay(300);
    digitalWrite(MB, LOW);
}
```

# ENVÍO DEL CÓDIGO

```
if(I2 = 1000) {
  Serial.println(I2);
  delay(1000);
  digitalWrite(LED_B, HIGH);
  digitalWrite(MB_B, HIGH);
  delay(500);
  digitalWrite(MB_B, LOW);
  digitalWrite(BB_B, HIGH);
  delay(500);
  digitalWrite(BB_B, LOW);
  digitalWrite(AB_B, HIGH);
  delay(400);
  digitalWrite(AB_B, LOW);
  digitalWrite(PC_B, HIGH);
  delay(1500);
  digitalWrite(PC_B, LOW);
  digitalWrite(AA_B, HIGH);
  delay(1700);
  digitalWrite(AA_B, LOW);
  digitalWrite(BA_B, HIGH);
  delay(1200);
  digitalWrite(HA_B, HIGH);
  digitalWrite(BA_B, LOW);
  delay(1500);
  digitalWrite(BB_B, HIGH);
  digitalWrite(HA_B, LOW);
  delay(1000);
  digitalWrite(BB_B, LOW);
}
```

```
if (readString == "1"){
  digitalWrite(LED_A, HIGH);
}
else if (readString == "2"){
  digitalWrite(LED_A, LOW);
}

if (readString == "3"){
  digitalWrite(LED_B, HIGH);
}
else if (readString == "4"){
  digitalWrite(LED_B, LOW);
}

readString = "";
}

void leer_dato() {
  while (Serial.available()) {
    delay(10);
    if (Serial.available() > 0) {
      dato = Serial.read();
      readString += dato;
    }
  }
}
```

# APLICACIÓN ANDROID



MIT App Inventor 2

ai2.appinventor.mit.edu/#6534395497283584

MIT App Inventor 2 Beta

Projects Connect Build Help My Projects Gallery Guide Report an Issue English israz1302@gmail.com

### hola\_mundo

Screen1 Add Screen ... Remove Screen Designer Blocks

#### Palette

User Interface

- Button
- TextBox
- ListView
- DatePicker
- TimePicker
- CheckBox
- Label
- ListPicker
- Slider
- PasswordTextBox
- Notifier
- Image
- WebView
- Spinner

Layout


Media

Drawing and Animation

Sensors

#### Viewer

Display hidden components in Viewer



#### Components

- Screen1

Rename Delete

#### Media

Upload File ...

#### Properties

Screen1

AboutScreen

AlignHorizontal: Left

AlignVertical: Top

AppName: hola\_mundo

BackgroundColor: White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Scrollable



# PANTALLA PORTADA



Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - TableArrangement1
    - Label9
    - menu
    - Image2
    - Image3
    - Image4
    - Label1
    - Image1
    - Label3

Rename Delete

Viewer







when menu .Click

do open another screen screenName "MENU"

# MENU

Guardando captura de pantalla...  Display hidden components in Viewer

MENU PRINCIPAL

 Control Brazo A	 Control Brazo B
 Control Automático	 Ayuda
 Acerca de	 Salir

20:54:28  
19/05/2015

Non-visible components  
BluetoothClient1 Clock1

```
when Control_A1.Click
do open another screen screenName "PORTADA"

when CONTROL_B1.Click
do open another screen screenName "CONTROL_B"

when AUTOMATICO.Click
do open another screen screenName "CONTROL_AUTO"

when OFF.Click
do close application

when AYUDA.Click
do open another screen screenName "AYUDA"

when ACERCA.Click
do open another screen screenName "acerca_d"
```

```
when Clock1.Timer
do set hora.Text to call Clock1.FormatTime instant call Clock1.Now
set fecha.Text to call Clock1.FormatDate instant call Clock1.Now
```

0 0  
Show Warnings



# BRAZO A



```
when Button13.Click
do close application

when DER.Click
do call BluetoothClient1.SendText
   text "A"

when UP_B.Click
do call BluetoothClient1.SendText
   text "C"

when UP_AB.Click
do call BluetoothClient1.SendText
   text "E"

when UP_M.Click
do call BluetoothClient1.SendText
   text "G"

when O_P.Click
do call BluetoothClient1.SendText
   text "H"

when PORTADA.Initialize
do if call BluetoothClient1.Connect
   address "30:14:06:06:10:08"
then set TableArrangement1.Visible to true
   set TableArrangement2.Visible to true

when IZQ.Click
do call BluetoothClient1.SendText
   text "B"

when DOWN_B.Click
do call BluetoothClient1.SendText
   text "D"

when DOWN_AB.Click
do call BluetoothClient1.SendText
   text "F"

when DOWN_M.Click
do call BluetoothClient1.SendText
   text "I"
```

# BRAZO B



```
when CONTROL_B.Initialize
do
  if BluetoothClient1.Connect address "30:14:06:06:10:08"
  then
    set TableArrangement1.Visible to true
    set TableArrangement2.Visible to true

when home.Click
do
  call BluetoothClient1.Disconnect
  open another screen screenName "MENU"

when close.Click
do
  close application

when der.Click
do
  call BluetoothClient1.SendText text "K"

when izq.Click
do
  call BluetoothClient1.SendText text "L"

when up_b.Click
do
  call BluetoothClient1.SendText text "M"

when down_b.Click
do
  call BluetoothClient1.SendText text "N"

when up_a.Click
do
  call BluetoothClient1.SendText text "A"

when down_a.Click
do
  call BluetoothClient1.SendText text "B"
```



# CONTROL AUTOMÁTICO



```
when CONTROL_AUTO .Initialize
do
  if BluetoothClient1 .Connect
    address "30:14:06:06:10:08"
  then
    set andro .Visible to true
    set Label3 .Visible to true
```

```
when ACTIVAR .Click
do
  call BluetoothClient1 .SendText
  text "8"
```

```
when Button4 .Click
do
  call BluetoothClient1 .Disconnect
  open another screen screenName "MENU"
```

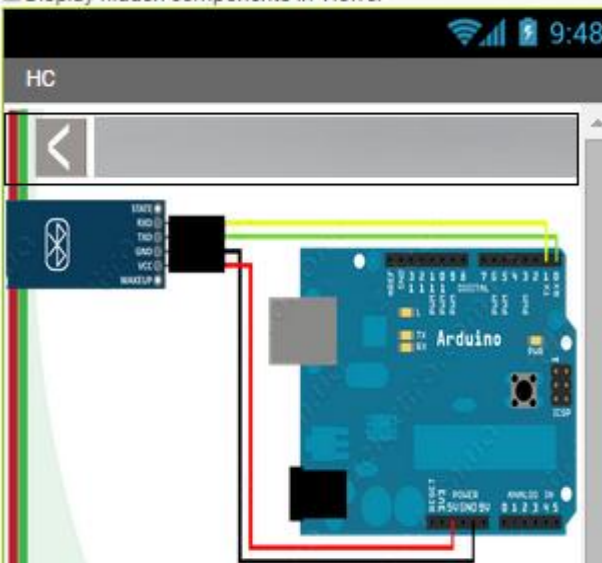
```
when Button5 .Click
do
  close application
```

```
when Iniciar .Click
do
  call BluetoothClient1 .SendText
  text "U"
```

# AYUDA

Display hidden components in Viewer

HC

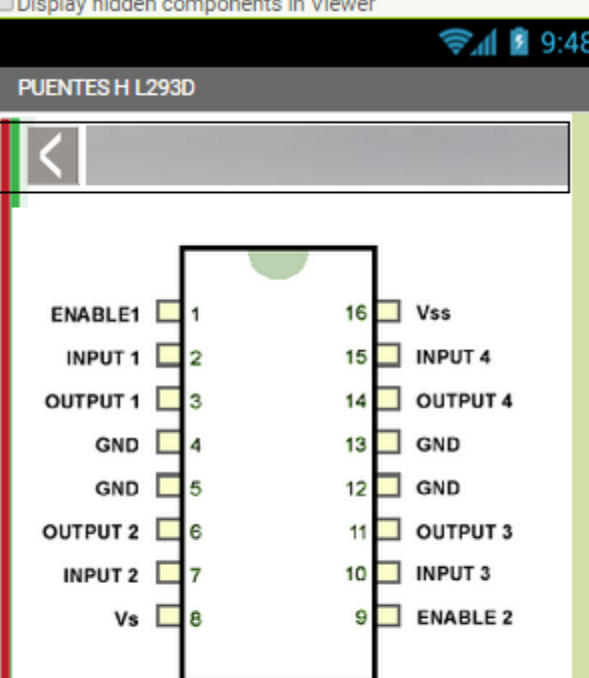


Para conectar el módulo es necesario cruzar la conexión entre la tarjeta y el HC06  
Es decir:  
Tx módulo - Rx Arduino  
Rx módulo - Tx Arduino

Nombre: TESIS\_IZ  
Contraseña: 1234

Display hidden components in Viewer

PUENTES H L293D



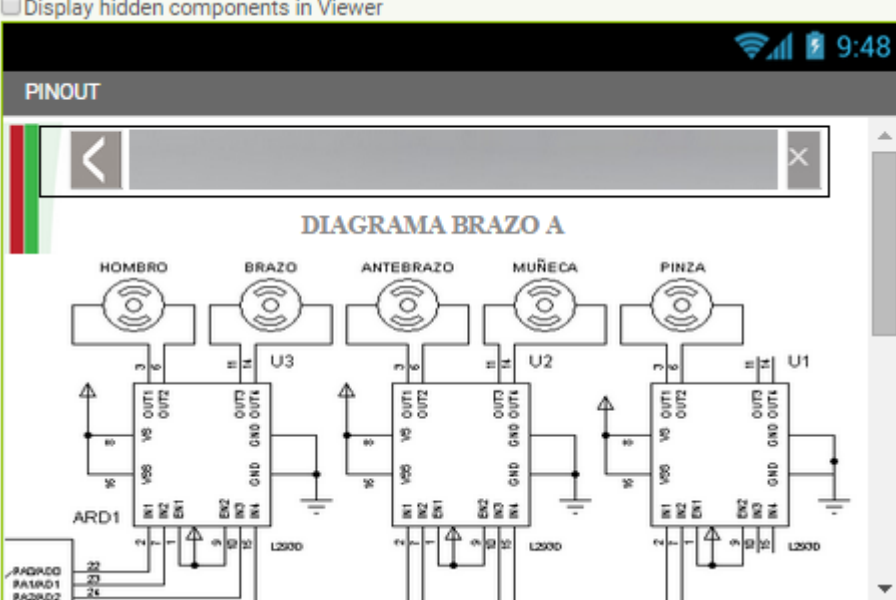
ENABLE1	1	16	Vss
INPUT 1	2	15	INPUT 4
OUTPUT 1	3	14	OUTPUT 4
GND	4	13	GND
GND	5	12	GND
OUTPUT 2	6	11	OUTPUT 3
INPUT 2	7	10	INPUT 3
Vs	8	9	ENABLE 2

- Rango de voltaje: 4.5 V a 36 V
- Protección ESD Interna
- Anagado térmico

Display hidden components in Viewer

PINOUT

DIAGRAMA BRAZO A



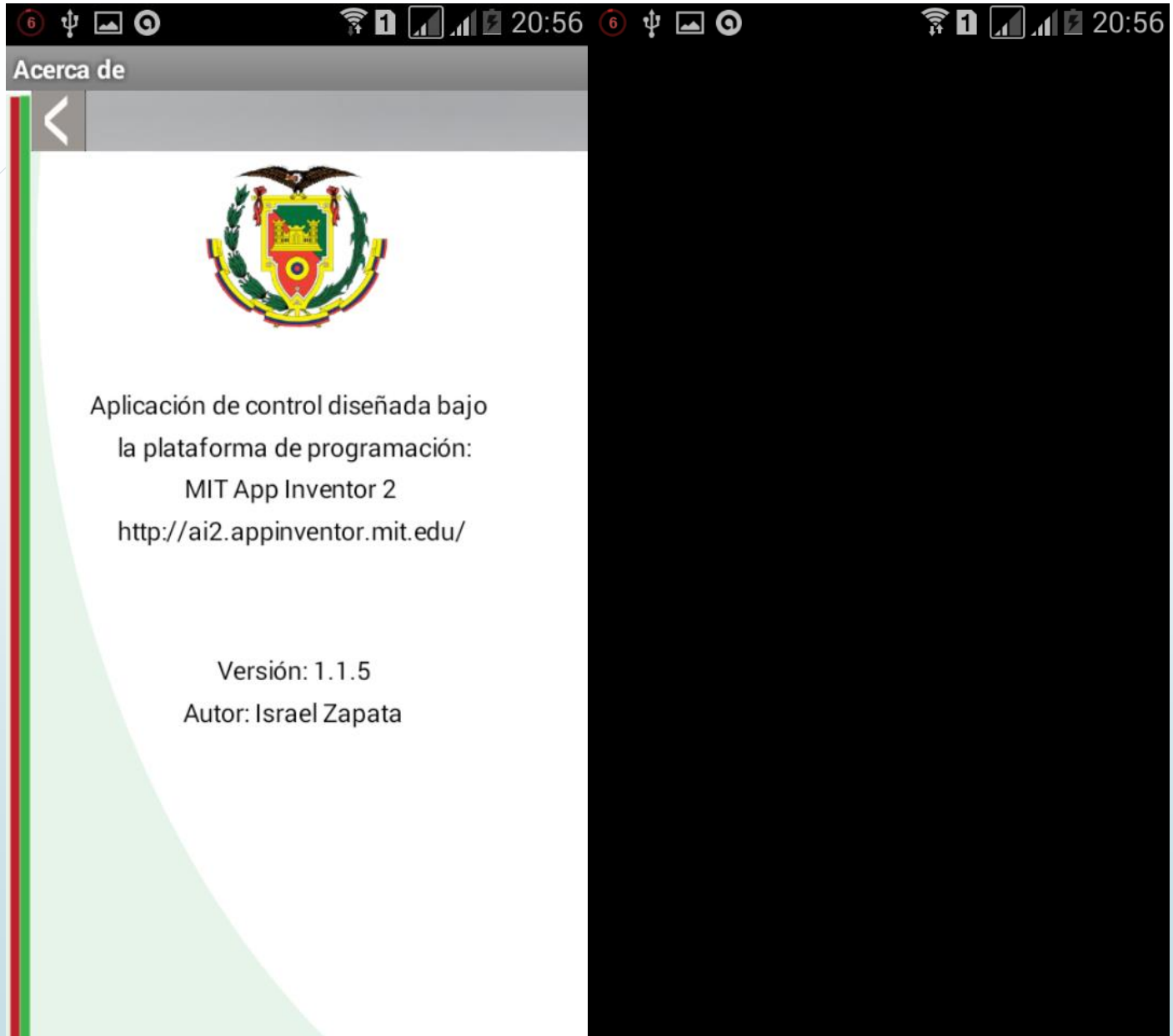
The diagram shows three L293D motor drivers (U1, U2, U3) connected to an Arduino (ARD1). Each driver controls a servo motor: U1 for the PINZA (gripper), U2 for the MUÑECA (wrist), and U3 for the BRAZO (arm). The HOMBRO (shoulder) and ANTEBRAZO (forearm) servos are also shown but not connected to the motor drivers in this specific diagram.

when Button1 .Click

do open another screen screenName "AYUDA"

when Button2 .Click

do close application





# SIMULACIÓN INDUSTRIAL





# CINTA TRANSPORTADORA



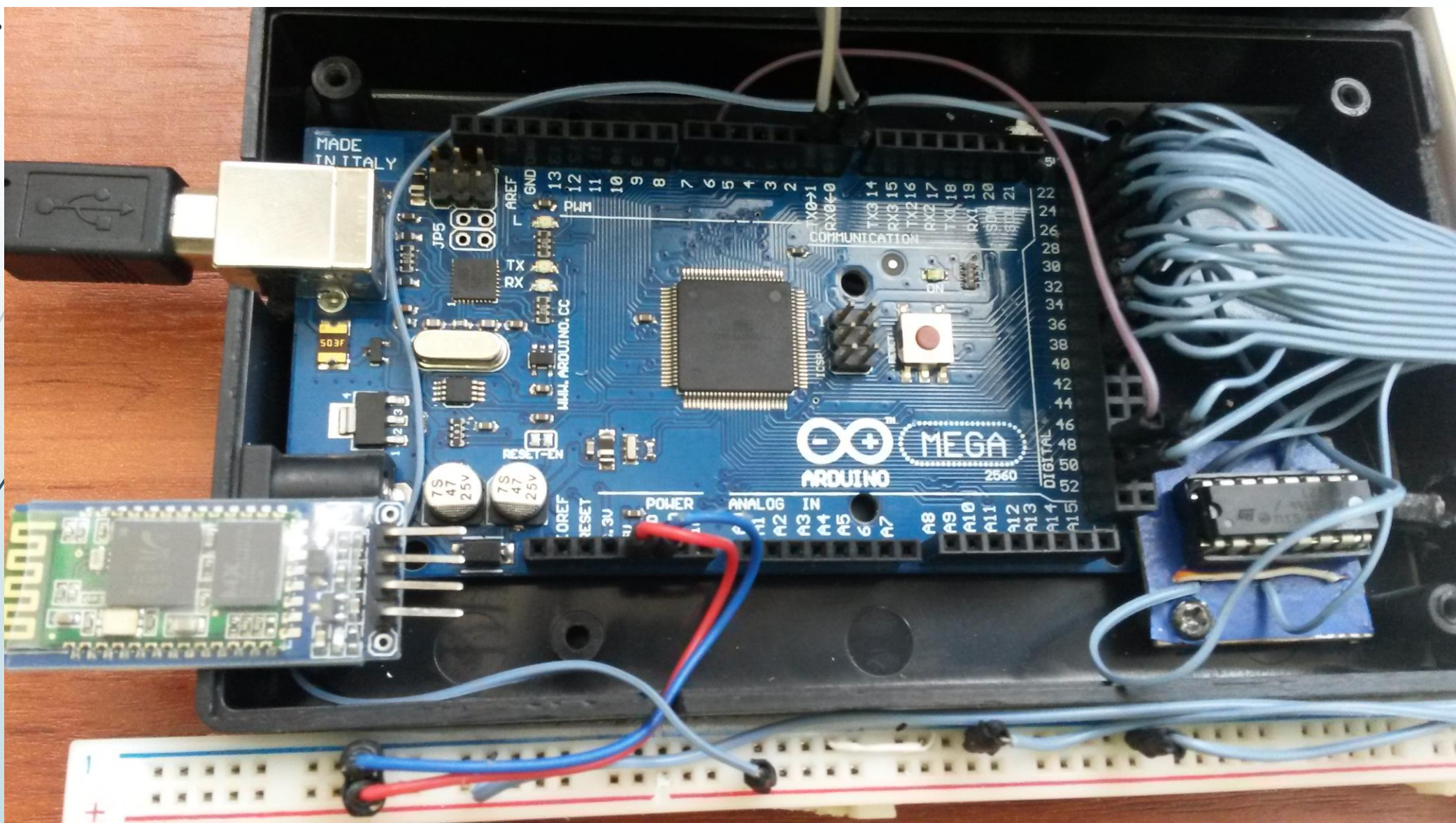
ISO9001 CE RoHS

1.5-24VDC  
1-1600rpm  
0.1-11kg.cm  
0.1-3.5W  
Dia. 12mm

[lunyeer.en.alibaba.com](http://lunyeer.en.alibaba.com)


**LUNYEE INDUSTRIES**





# CONCLUSIONES

- Posterior a un análisis detallado de los tipos de tarjetas Arduino, se pudo seleccionar a la adecuada para esta aplicación, tomando en cuenta detalles como el número de puertos disponibles en cada tarjeta y la capacidad de su procesador central.
- La comunicación bluetooth facilita el control inalámbrico de procesos debido a la velocidad de su comunicación, y dicha comunicación se realizó de manera correcta y efectiva enlazando el módulo HC 06 y la tarjeta Arduino mediante líneas de programación básicas estableciendo así una velocidad de transmisión de 9600bps.


- 
- ▶ El desarrollo de la aplicación para el sistema operativo Android permite evidenciar que este lenguaje de programación es cada vez más fácil de aprender y sobre todo capaz de cumplir con casi cualquier requerimiento de control adquisición y manipulación de variables de cualquier tipo, de esta manera se hace posible una adaptación a cada necesidad de automatización y control requerido por un estudiante.
  - ▶ El prototipo de proceso industrial sin duda es el medio de interacción perfecto para poner a prueba a los brazos guiados por la aplicación móvil instalada en el Smartphone, recalcando también que se han realizado pruebas instalando en diferentes dispositivos móviles de SO Android la aplicación de control.



# RECOMENDACIONES

- ▶ Trabajar cuidadosamente con los voltajes de alimentación de los brazos robóticos, de la tarjeta Arduino y de los integrados de control de giro L293D, ya que para un mejor resultado sería mejor alimentar a los brazos con su fuente independiente del sistema, es decir con cuatro baterías tipo D, se evidenció que al alimentarlos con la fuente que incorpora Arduino existe un corto circuito en una de las salidas y es posible incluso llegar a quemar los motores de los ejes del robot o incluso la misma tarjeta. Los demás componentes si pueden ser alimentados con la fuente de 5V de Arduino y el módulo Bluetooth incluso puede ser alimentado con la fuente de 3V.
- ▶ Tomar en consideración el alcance y las limitaciones que tiene MIT App Inventor ya que este software en su aspecto visual es muy delimitante y se debe optar por implementar varias imágenes para mejorar la interfaz de la aplicación, es recomendable utilizar programas de edición de imágenes especializados de tal manera que mejore aún más la estética de la aplicación. Además que es un desarrollador exclusivo para Android.



- 
- ▶ Verificar la conexión entre las entradas y las salidas de la tarjeta Arduino hacia los integrados L293D y de éstos hacia los actuadores del robot, ya que si el sentido de giro está contrario al especificado es porque dos de las entradas o salidas están conectadas erróneamente, la solución más fácil sería intercambiar físicamente los cables de comunicación.
  - ▶ Tener muy presente que aunque el trabajo investigativo se desarrolló en base a una tarjeta Arduino Mega esto no quiere decir que sea una aplicación de uso exclusivo por esta tarjeta, en el caso de querer desarrollarla sobre una tarjeta Arduino diferente, es necesario cambiar la programación Arduino específicamente en la asignación de entradas y salidas, tomando en cuenta la limitación de puertos de cada tarjeta. En la aplicación no hace falta ningún otro cambio o modificación.



**GRACIAS POR SU AMABLE  
ATENCIÓN**