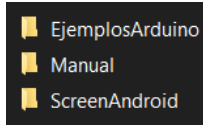


Tutorial Para conectar Arduino a Pantalla Android

Paso 1 Download library and install

Descarga o clona el repositorio en: <https://github.com/johnspice/libraryScreenArduino>

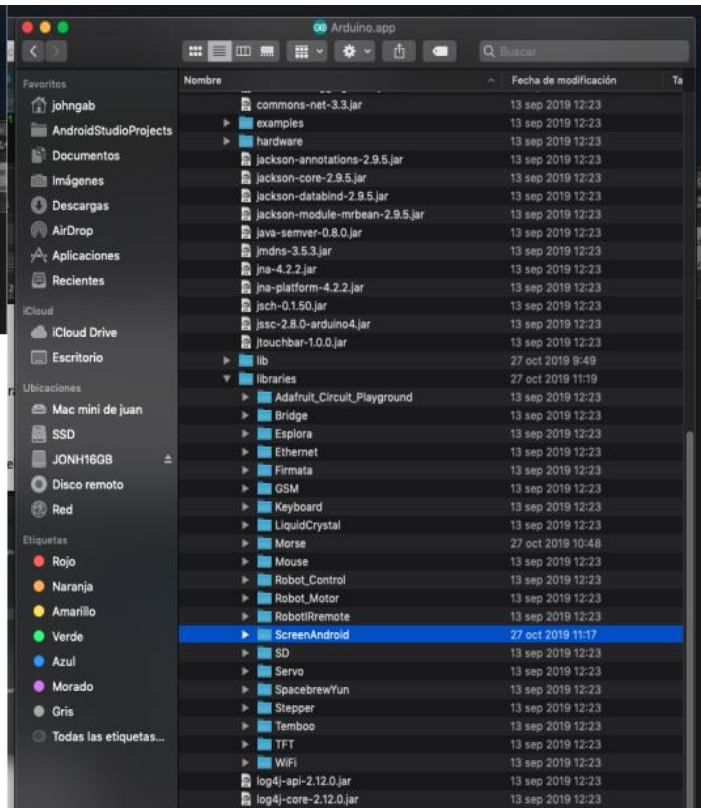
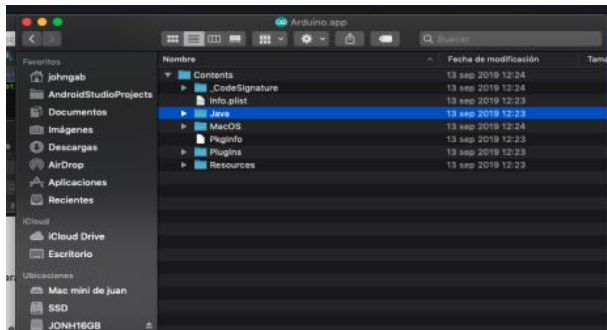
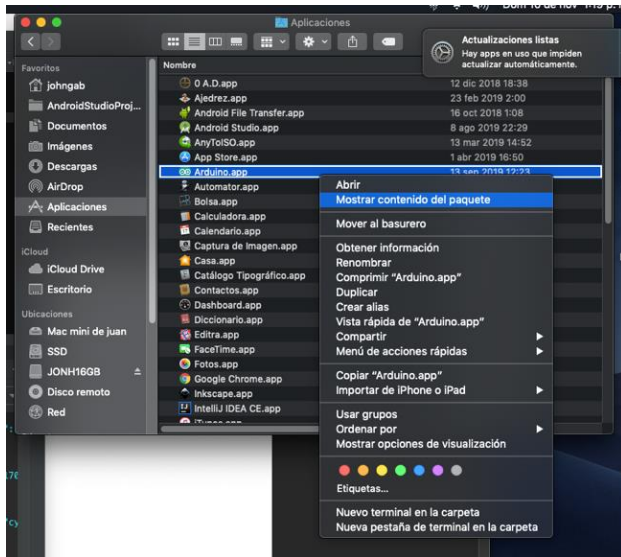
Se descargarán 3 carpetas:



Copy "ScreenAndroid" into "libraries" from arduino:

En MAC

Aplicaciones/arduino.app/java/libraries

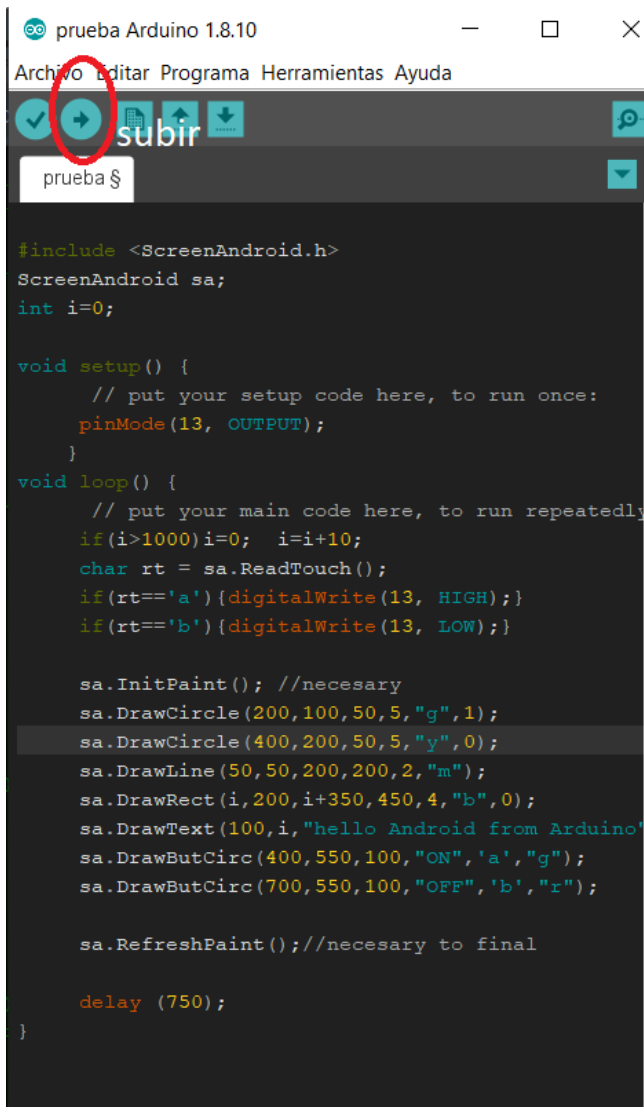


En Windows

Copy into C:\files Program (x86)\Arduino\libraries

Paso 2 Run Test Code

Abrir arduino y cargar el código “prueba.ino” de prueba que está dentro de la carpeta “EjemplosArduino”. Elegir tarjeta puerto serie y subir a la placa. (en la siguiente sección se explica el código)



```
#include <ScreenAndroid.h>
ScreenAndroid sa;
int i=0;

void setup() {
    // put your setup code here, to run once:
    pinMode(13, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly
    if(i>1000)i=0; i=i+10;
    char rt = sa.ReadTouch();
    if(rt=='a'){digitalWrite(13, HIGH);}
    if(rt=='b'){digitalWrite(13, LOW);}

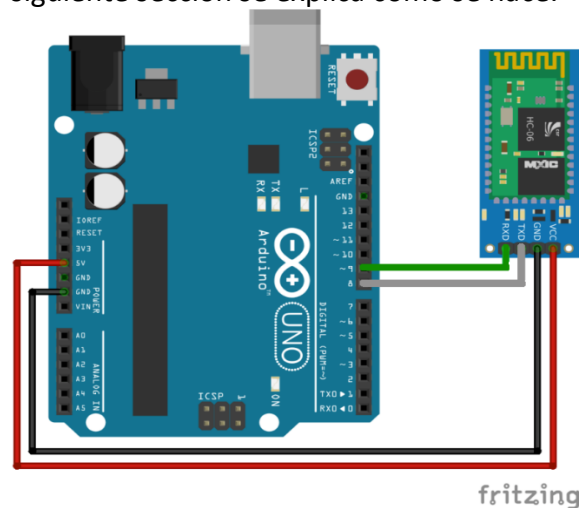
    sa.InitPaint(); //necesary
    sa.DrawCircle(200,100,50,5,"g",1);
    sa.DrawCircle(400,200,50,5,"y",0);
    sa.DrawLine(50,50,200,200,2,"m");
    sa.DrawRect(i,200,i+350,450,4,"b",0);
    sa.DrawText(100,i,"hello Android from Arduino");
    sa.DrawButCirc(400,550,100,"ON",'a',"g");
    sa.DrawButCirc(700,550,100,"OFF",'b',"r");

    sa.RefreshPaint();//necesary to final

    delay (750);
}
```

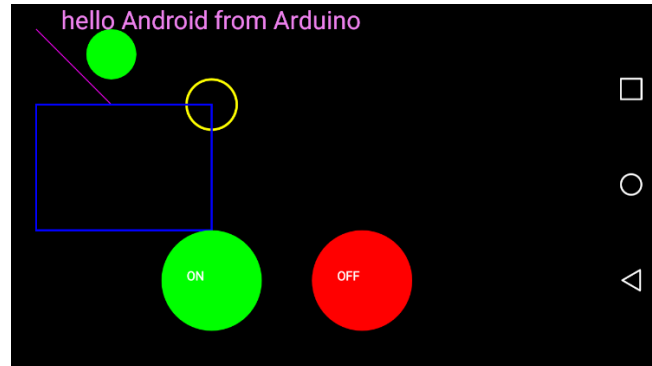
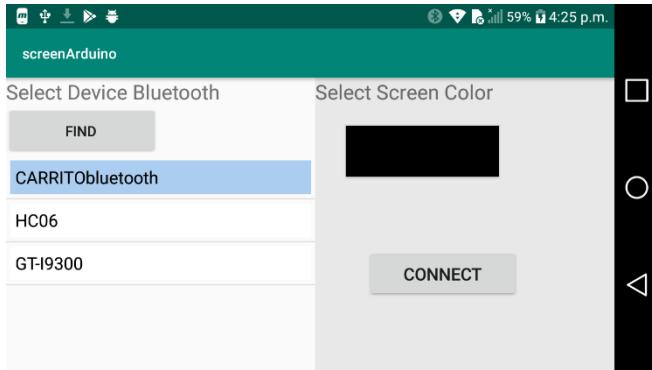
Paso 3 Conexión Arduino-bluetooth (hc-05 o hc06)

Por default la conexión es en los pines rx=8 y tx= 9, a 9600 baudios. Como se muestra a continuacion. Si se requiere pintar muchos elementos en pantalla y tener un tiempo de refresh menor a 750ms se debe cambiar este valor al maximo a 115200, tenga en cuenta que cambiarlo hara necesario cambiarlo tambien en los hc05/06 usando comandos At. En la siguiente seccion se explica como se hace.



Paso 4 Conexión hc05/06-Android

- Ir a ajustes en android y vincular el bluetooth hc06/05
- open App in android, click find, select bluetooth, seleccionar color de fondo de la pantalla , connect.



Deberia poderse ver algo como lo anterior.

Descripción del API ScreenAdroid

Explicacion de uso y funciones incluidas dentro de la librería ScreenArduino

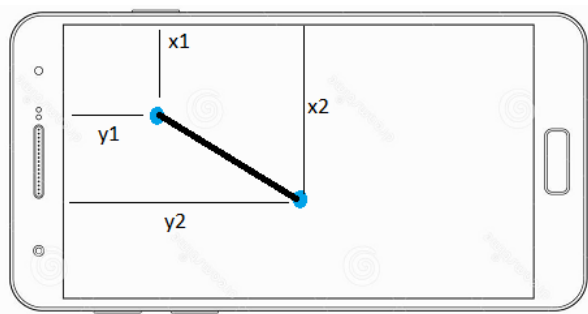
Antes de comenzar a pintar es necesario llamar al metodo **InitPaint();** y al finalizar se debe de llamar al metodo **RefreshPaint();**

Si no se colocan esos dos metodos provocara errores an android

La librería dispone de solo algunos colores basicos que seran llamados con iniciales

Dibujar lineas

DrawLine(int x1,int y1, int x2,int y2, int w,string color);



(x1,y1) punto inicial en pixeles

(x2,y2) punto final en pixeles

w=grueso del trazo en pixeles

color =color de la linea solo existen los siguientes valores :green=g,yellow=y,orange=o,red=r, violet=v,blue=b,cyan=c, magenta=m,white=w,black=bc,gray=gr

ejemplo: DrawLine(50,50,200,200,2,"m"); el color de la linea seria magenta

Dibujar circulo

DrawCircle(int x,int y,int r,int w,string color,int fill);

Centro (x,y) en pixeles

r=radio en pixeles

w=grueso del trazo

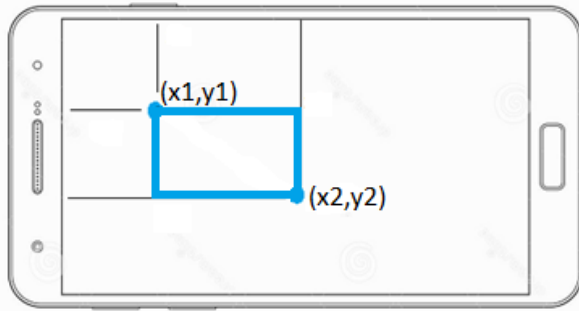
color=mismos colores que la linea

fill= puede ser 1 ó 0 1=circulo relleno, 0=solo contorno.

Ejemplo:DrawCircle(200,100,50,5,"g",1); circulo color verde relleno

Dibujar rectangulo

DrawRect(int x1,int y1,int x2,int y2,int w,int color,int fill);



color = mismos que linea

Si fill=1 el rectangulo se rellena fill=0 solo se traza el contorno

Ejemplo: DrawRect(100,200,350,450,4,"b",0);

Dibujar Texto

DrawText(int x,int y,string mensaje,int size,string color);

Posicion del texto (x,y),mensaje=texto a imprimir, size tamaño del texto debe ser un valor entre 1 y 20.

Color=los mismo que linea

Ejemplo; DrawText(100,200,"hello Android from Arduino",5,"v");

Dibujar boton Circular

DrawButCirc(int x,int y,int r,string texto,char return,string color);

Centro del boton (x,y) radio=r.

return=es el carácter que se enviara desde android de forma serial al tocar el boton sobre la pantalla

color= los mismos que linea, el boton siempre es relleno.

ejemplo

DrawButCirc(700,550,100,"OFF",'b',"r");

Se dibujara un botn circular en la posicion x=700,y=550 r=100 todo en pixeles. Con texto off

Al tocar el boton se enviara hacia arduino via serie el carácter 'b', el boton es de color rojo.



El valor enviado se cacha en arduino usando el metodo: **ReadTouch();**

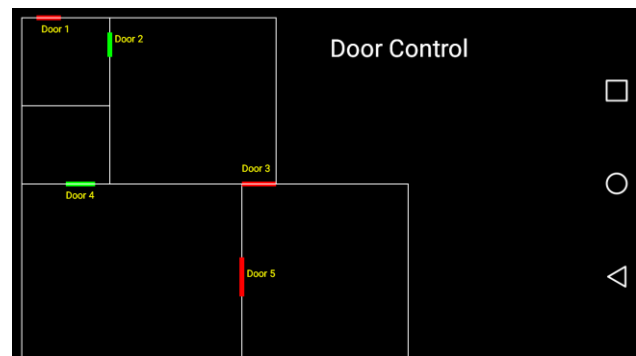
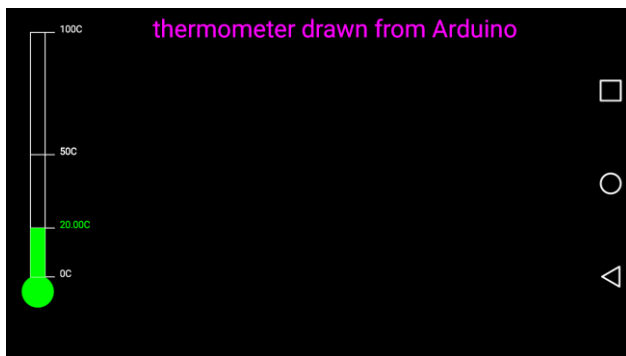
Ejemplo:

En el codigo de prueba se tiene el siguiente bloque de codigo

```
char rt = sa.ReadTouch();
  if(rt=='a'){digitalWrite(13, HIGH);}
  if(rt=='b'){digitalWrite(13, LOW);}
```

si se preciona el boton verde “on” se envia ‘a’ y se prende el led del arduino uno del pin 13, si se preciona el boton rojo “off” se apaga el led.

Se incluyen 2 ejemplos mas de como pintar termometro (ejemplo1.ino) y un plano de una casa (ejemplo2.ino)muy simples.



Nota: ScreenArduino funciona bien para pocos elementos dibujados en pantalla (unos 15 elementos: líneas círculos...) con refresh (delay) mayores a 1seg.

Si el delay en arduino es mayor a 1 segundo la velocidad de reacción del botón circular sufrirá un retraso al click. Si se desea dibujar muchos elementos con refresh de 100ms se debe cambiar la velocidad a 115200baudios en el archivo “ScreenAndroid.cpp” que se copió a libraries dentro de arduino.

La librería se probó con arduino uno, mega y Nodemcu Modulo Wifi Esp8266, aunque este último es necesario cambiar los pines y el led de prueba a la librería para arduino.