Arduino in Physical Etoys

Due to the popularity and the different features that the Arduino board has, in this tutorial we will see how to access to some of them from Physical Etoys in order to make simple programs and have a suitable introduction to electronics.

Necessary tools:

1. An Arduino board (we have worked with a Duemilanove).
2. An USB cable.
3. Physical Etoys software.

Optional tools:

1. A led, a pushbutton, a protoboard, 3 10 KOhms resistors and a servomotor (to prove that the board works).

Driver installation:

When you connect the board, Windows should initiate the driver installation process (if you haven't used the computer with an Arduino board before).

On Windows Vista, the driver should be automatically downloaded and installed.

On Windows XP, the Add New Hardware wizard will open:

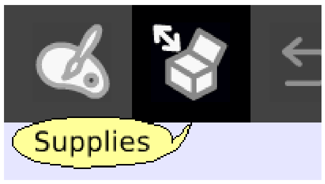
- When asked: Can Windows connect to Windows Update to search for software? Select No, not this time. Click next.
- Select Install from a list or specified location (Advanced) and click next.
- Make sure that Search for the best driver in these locations is checked; uncheck Search removable media; check "Include this location in the search" and browse the drivers in the /FTDI USB Drivers directory of the Arduino distribution. (The latest version of the drivers can be found on the FTDI website). Click next.
- The wizard will search for the driver and then tell you that a "USB Serial Converter" was found. Click finish.
- The new hardware wizard will appear again. Go through the same steps and select the same options and location to search. This time, a "USB Serial Port" will be found.

You can check that the drivers have been installed by opening the Windows Device Manager (in the Hardware tab of System control panel). Look for a "USB Serial Port" in the Ports section; that's the Arduino board.

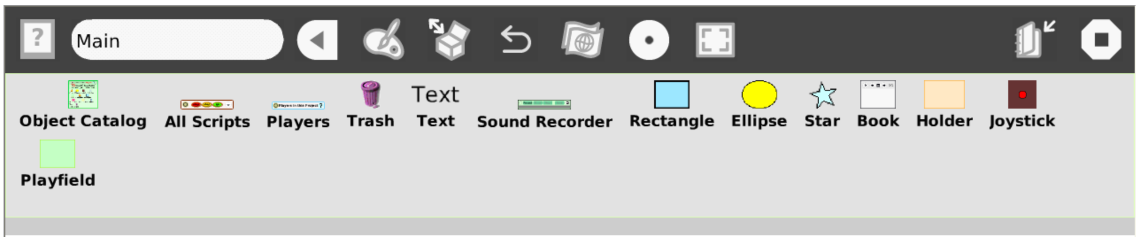
Connecting Arduino to Physical Etoys:

First and foremost, inside Physical Etoys, we have to get an “Arduino board”. This is a graphical object that represents the Arduino board.

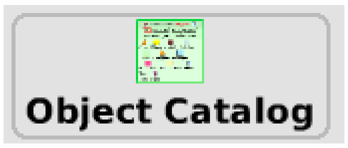
In order to obtain it we have to open the supplies’ flap.



The supplies’ flap contains the most used objects. As long as we use the system we are going to learn more things about them. The one that interests us is the “Object catalog”.

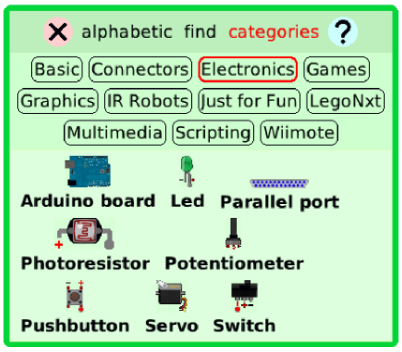


Now we have to drop the object catalog on the world.



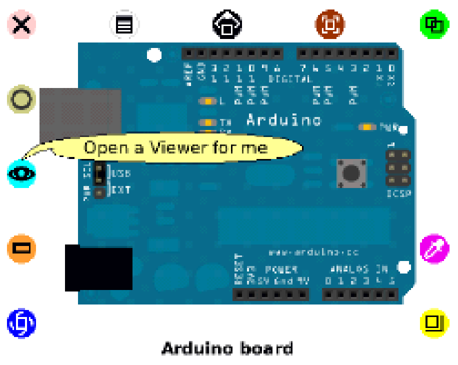
The object catalog is like a box that contains the entire objects that we can use. It is ordered by categories but we can arrange the objects alphabetically. Apart from that we can look for a particular one.

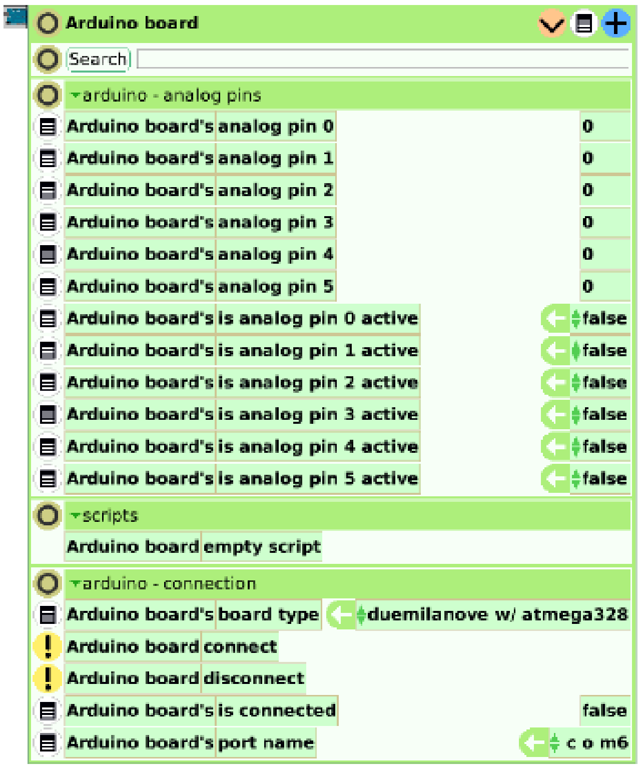
Now we have to choose the Electronics category. Then we have to drag the “Arduino board” etoy and place it on the world.



Now we have to open its viewer by making right-click on the “Arduino board” to open the halo. The halo is a set of buttons which surround the object and let the user modify, move, delete and maximize it.

Next we click on the viewer icon (light blue) to make it appear. The viewer is a flap where we can not only see and modify the object’s properties on the screen but also create scripts for them to perform actions like moving on the screen. The properties and the actions are represented as tiles.




Arduino board

Search

arduino - analog pins

Arduino board's analog pin 0	0
Arduino board's analog pin 1	0
Arduino board's analog pin 2	0
Arduino board's analog pin 3	0
Arduino board's analog pin 4	0
Arduino board's analog pin 5	0
Arduino board's is analog pin 0 active	false
Arduino board's is analog pin 1 active	false
Arduino board's is analog pin 2 active	false
Arduino board's is analog pin 3 active	false
Arduino board's is analog pin 4 active	false
Arduino board's is analog pin 5 active	false

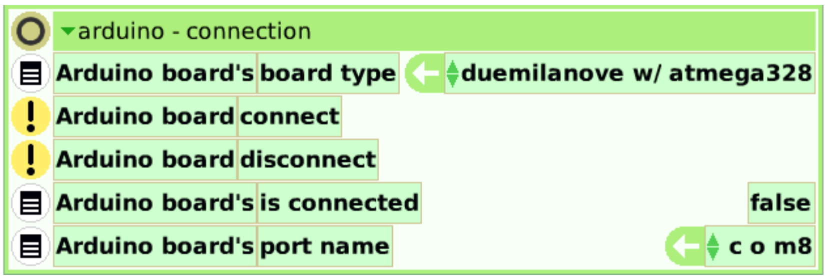
scripts

Arduino board empty script

arduino - connection

Arduino board's board type	duemilanove w/ atmega328
Arduino board connect	
Arduino board disconnect	
Arduino board's is connected	false
Arduino board's port name	c o m6

In a special category called “arduino-connection” we can find instructions which are useful for connecting with the Arduino board. In order to change categories we have to double click the tiny triangle which is on the left hand side of the title of the categories and then we have to choose the desired category.



arduino - connection

Arduino board's board type ← duemilanove w/ atmega328

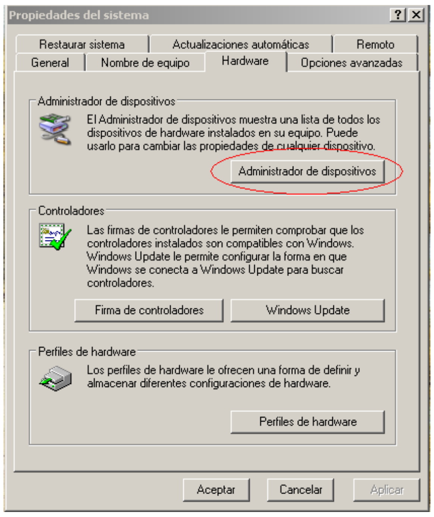
Arduino board connect

Arduino board disconnect

Arduino board's is connected false

Arduino board's port name ← c o m8

First we have to specify which board we are going to use. In this case we will choose the Duemilanove board with the ATmega328 microcontroller. Then we have to define the COM port the board is connected. To know where is connected, we have to right-click in My computer→properties. Next, inside the hardware tab we have to click on the device manager.



Propiedades del sistema

Restaurar sistema | Actualizaciones automáticas | Remoto
General | Nombre de equipo | Hardware | Opciones avanzadas

Administrador de dispositivos

El Administrador de dispositivos muestra una lista de todos los dispositivos de hardware instalados en su equipo. Puede usarlo para cambiar las propiedades de cualquier dispositivo.

Administrador de dispositivos

Controladores

Las firmas de controladores le permiten comprobar que los controladores instalados son compatibles con Windows. Windows Update le permite configurar la forma en que Windows se conecta a Windows Update para buscar controladores.

Firma de controladores | Windows Update

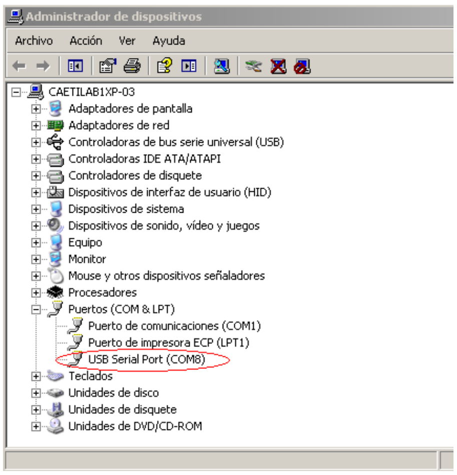
Perfiles de hardware

Los perfiles de hardware le ofrecen una forma de definir y almacenar diferentes configuraciones de hardware.

Perfiles de hardware

Aceptar | Cancelar | Aplicar

Now we have to look for the "ports" section and once inside, the port number will be next to the label "USB Serial Port". In this case the port is 8.

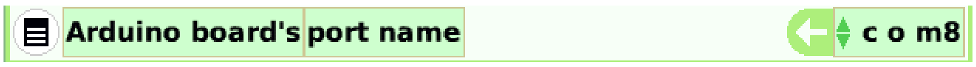


Administrador de dispositivos

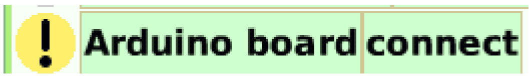
Archivo Acción Ver Ayuda

```
CAETILAB1XP-03
  Adaptadores de pantalla
  Adaptadores de red
  Controladoras de bus serie universal (USB)
  Controladoras IDE ATA/ATAPI
  Controladores de disquete
  Dispositivos de interfaz de usuario (HID)
  Dispositivos de sistema
  Dispositivos de sonido, vídeo y juegos
  Equipo
  Monitor
  Mouse y otros dispositivos señaladores
  Procesadores
  Puertos (COM & LPT)
    Puerto de comunicaciones (COM1)
    Puerto de impresora ECP (LPT1)
    USB Serial Port (COM8)
  Teclados
  Unidades de disco
  Unidades de disquete
  Unidades de DVD/CD-ROM
```

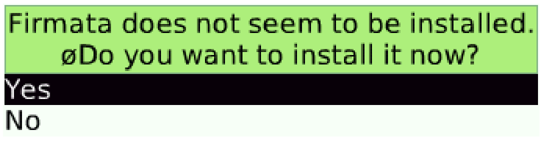
Therefore, the value of the tile which corresponds to the port will be COM 8


Arduino board's port name c o m8

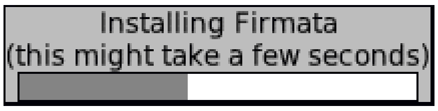
Now the important instruction is “connect”. If we run this instruction (by clicking on the yellow button with an exclamation sign) we will connect the Arduino with Physical Etoys and then the tile which indicates if the Arduino board is connected will change its value to true.


Arduino board connect

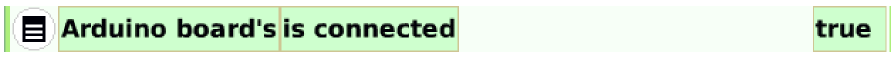
If this is the first time that we have tried to connect the Arduino board, a pop up will appear asking if you want to install Firmata. Firmata is a program which contains the communication protocol to enable the board to receive orders. So, we click on the yes button.


Firmata does not seem to be installed.
øDo you want to install it now?
Yes
No

A bar will show the installation progress.

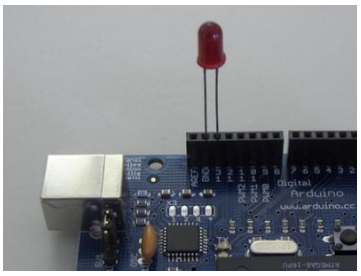

Installing Firmata
(this might take a few seconds)

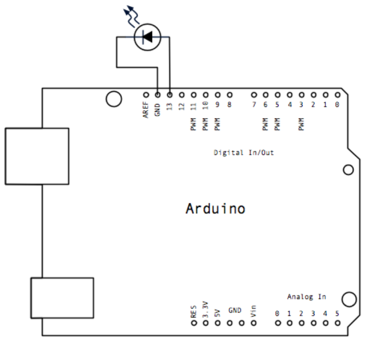
If no error occurred, a caption will appear indicating that the process ended correctly. Now we have to connect the Arduino board again (by clicking on the exclamation sign of the connect tile). If the Arduino board is connected, the value of the isConnected tile will become true.


Arduino board's is connected true

The “hello world” of electronics (blinking a led)

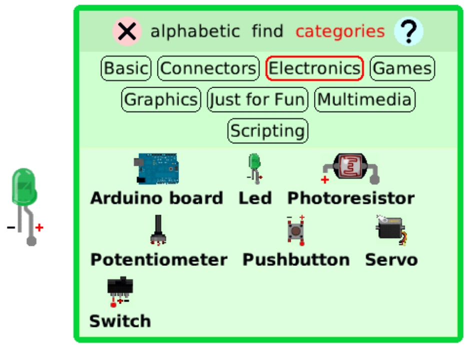
Before working with Physical Etoys we need to connect physically the led with the board. In this example, the cathode must be connected to **ground** and the anode to the **thirteenth pin**.



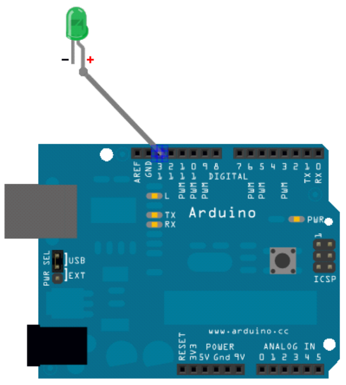


Now we can start with Physical Etoys.

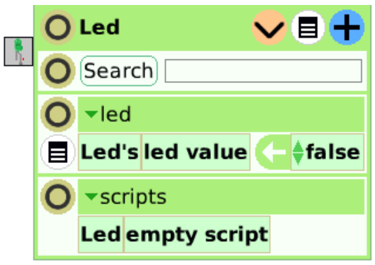
Inside the “electronics category” from the object catalog, we drag a led.



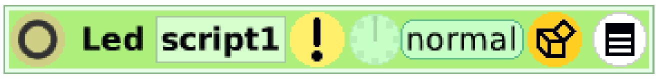
Once dropped, we have click on the positive pin and drag the appearing cable to one of the digital pins of the “Arduino board” (in this case the thirteen pin).



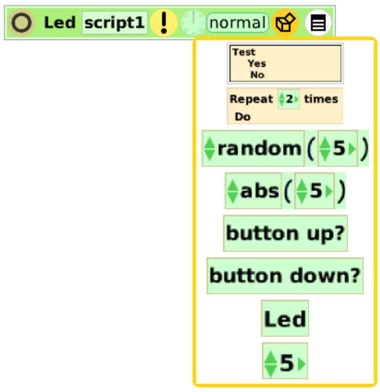
Next we go to the led’s viewer and these tiles will appear:



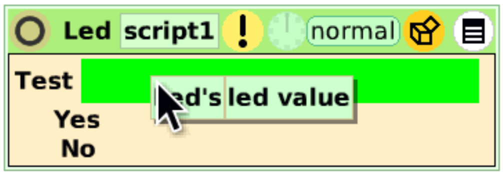
Now we have to create a new script to make the led blinks. To create a new script we have to drag the “empty script” tile which is inside the “scripts” category. Once dragged we will see this:



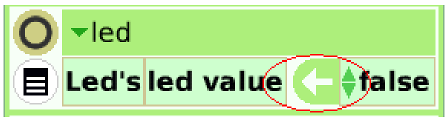
Then we have to click on the box button and choose a “test yes/no” tile. This tile means that the script will do something according to a condition (in this case the led’s value). For the ones who have a bit of experience in programming this tile is an “if structure”.

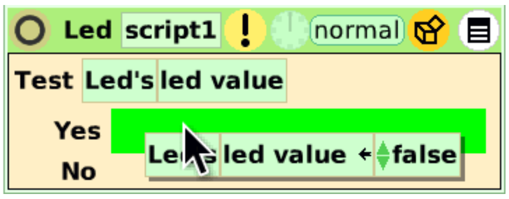


In order to establish the condition, we will drag (not from the arrow because we do not want to assign values to variables) the “led value tile” next to “test”:

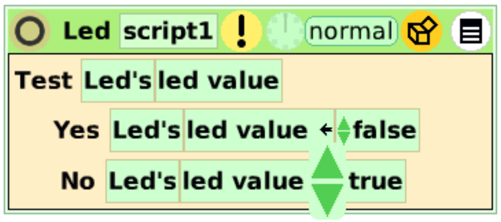


Now we have to define what the script will do if the led is turned on. Obviously, it has to turn it off. So we will assign to “led value” a false value. So we have to drag the led value from the arrow because we want to assign a value to the variable.

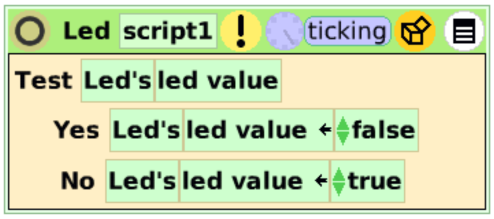




Now we will do the same but with a subtle difference: when the led is off, Physical Etoys will turn it on. So we have to drag from the arrow the “led value” tile to the “No” section of the “Yes/No tile” but we will change its value to true.



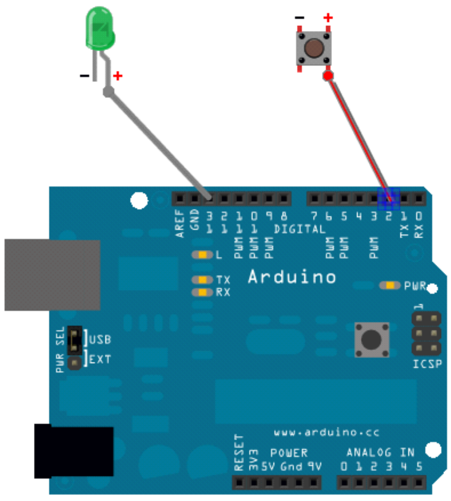
Finally we will run the script by clicking on the clock icon.



If everything was fine, the led will blink.

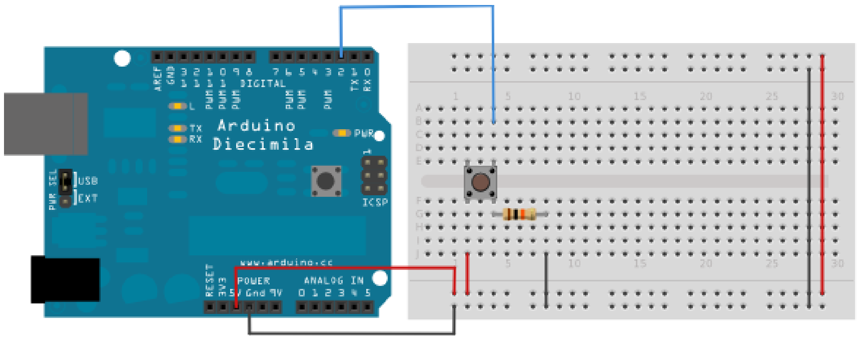
Turning on a led by pressing a button

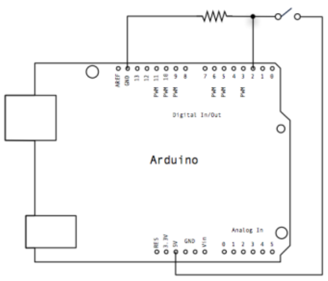
Inside the Electronics category we have to drag a “push button” and then we have to connect it to the second pin of the “Arduino board”.



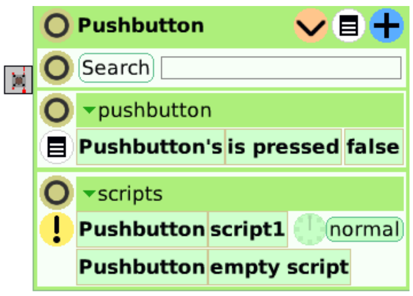
Take into account not to disconnect the led from the thirteen pin. Apart from that, stop the led's script because it will interfere with the objective of this exercise.

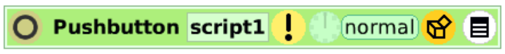
The physical connection of the button (with the resistor and the protoboard) should be like this:



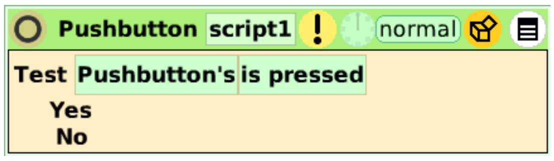


Now we have to enter to the Pushbutton's viewer and create a new script.

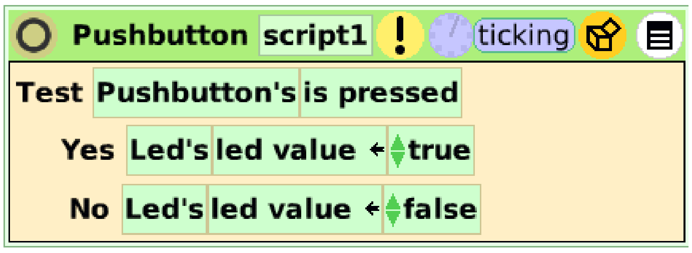

Pushbutton
Search
pushbutton
Pushbutton's is pressed false
scripts
Pushbutton script1 normal
Pushbutton empty script


Pushbutton script1 normal

Then we add a "Test yes/no" tile to know when the button is pressed. We have to drag the tile from its name because it is not an assignment. We want to consult its value.


```
Pushbutton script1 normal
Test Pushbutton's is pressed
  Yes
  No
```

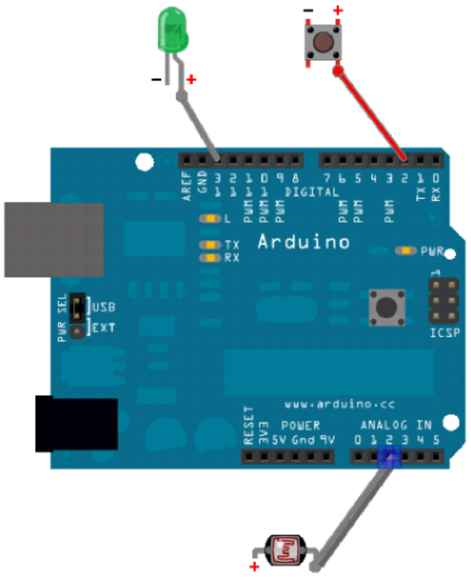
In case the button is pressed we will assign to the led a true value, otherwise we will assign a false value. Remember to drag the tiles from the arrows. Finally we have to click on the clock to run the script.


```
Pushbutton script1 ticking
Test Pushbutton's is pressed
  Yes Led's led value ← true
  No Led's led value ← false
```

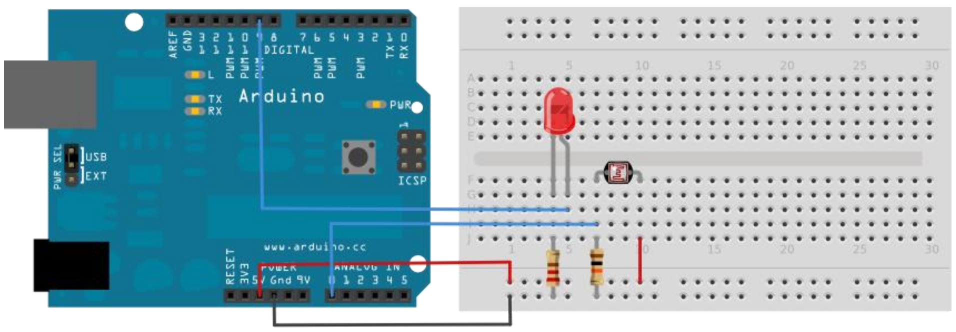
If everything was fine, when we press the button, the led will blink.

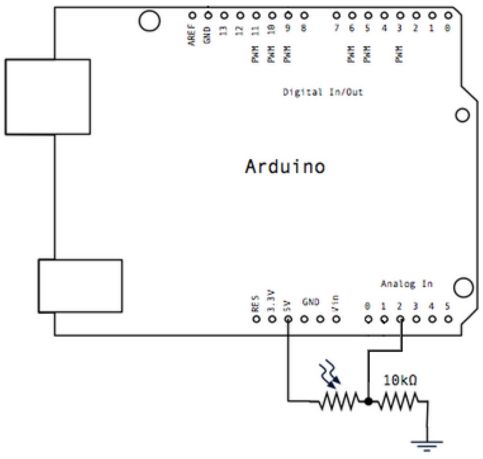
Reading a Photoresistor

As we have the "Arduino Board" connected, we will attach a Photoresistor (which is inside the electronics category) to the second analog input pin.

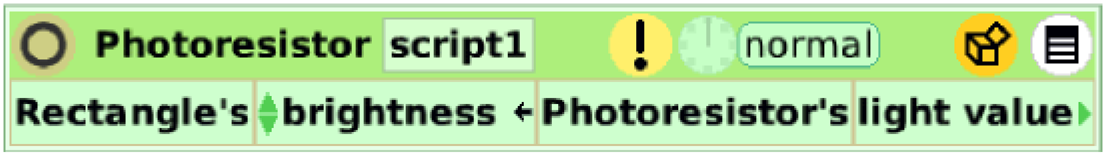


The real connection should be more or less like this:





Next we open the photoresistor's viewer and create a new script. Now we have to go to the object catalog and drag from the basic category a rectangle. Then we open the rectangle's viewer in order to drag (from the arrow) the brightness tile inside the photoresistor's script. Now we have to open the photoresistor's viewer to replace the rectangle's brightness value (100) with the "Photoresistor's light value" tile (not from the arrow because we only want to read the value).



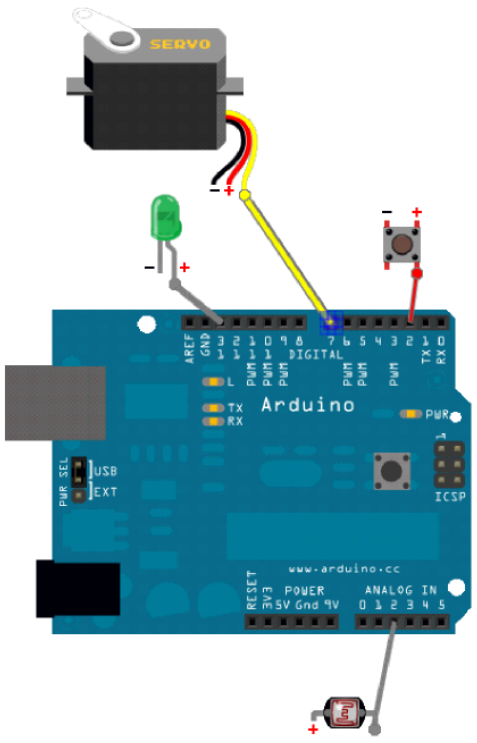
Photoresistor script1 normal

Rectangle's brightness ← Photoresistor's light value

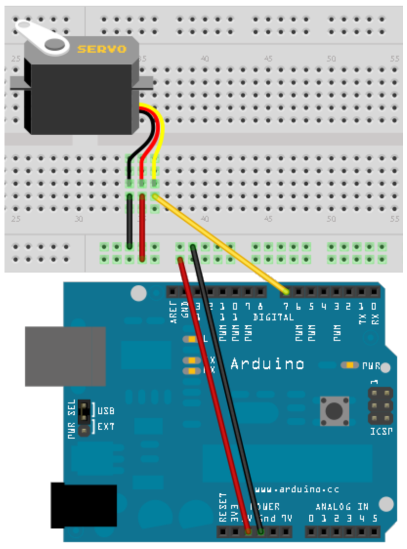
Finally we run the script to see how the rectangle changes its brightness according to the ambient light.

Rotate a Servo with a rectangle

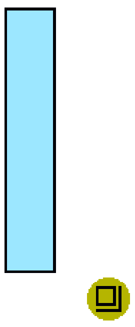
As we have the “Arduino Board” connected, we will attach a Servo (which is inside the electronics category) to the seventh pin.



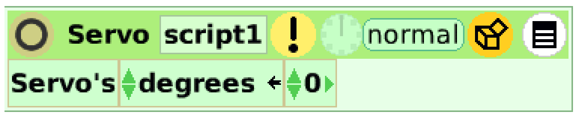
The physical connection should be like this:



In order to save time we can use the rectangle created in the last exercise. We open its halo and then we drag the yellow button to change its size.



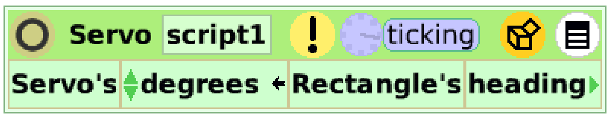
Now we have to open the servo's viewer and create a new script with the tile that contains the servo's degrees. Remember that it is an assignment; therefore, we will drag the tile from the arrow.



Servo script1 normal

Servo's degrees ← 0

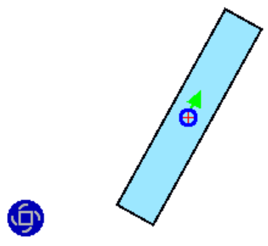
Then we have to open the rectangle's viewer and inside the basic category we have to drag (from its value) the "heading" tile. Next we run the script.



Servo script1 ticking

Servo's degrees ← Rectangle's heading

While the script is running we have to open the rectangle's halo. Finally we have to drag the blue button to rotate the rectangle.



If everything was fine the motor will move according to the rectangle’s heading.

Conclusion

Well, that is basically all that we need to begin using the Arduino board. The possibilities of interaction between the computer, Arduino and the other electronic components that Physical Etoys provides are very numerous to deal with everyone in this small tutorial. We encourage you to discover the other ones by exploring the environment (testing, playing, touching and breaking if it is necessary)

Have fun!