
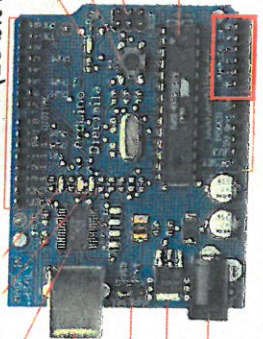



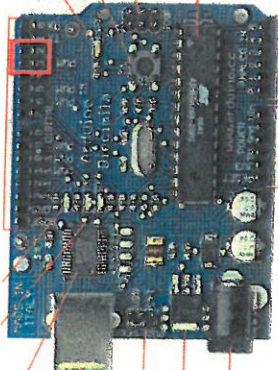




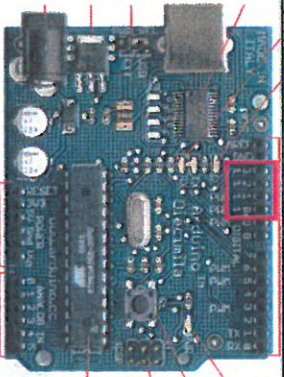

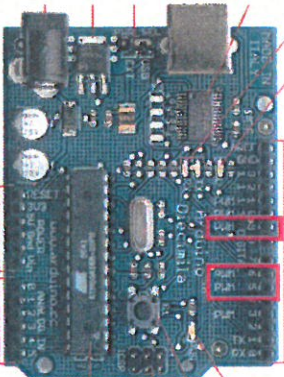


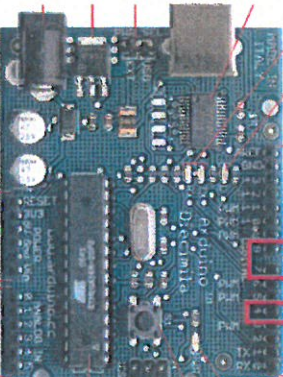



Scratch for Arduino Cheat sheet

<u>INPUTS</u>		<u>USEFUL FOR:</u>
	<p>Read A0 through A5... stores a number between 0 and 1023 (10 bit number)</p> 	<p>Light Dependent Resistor (LDR)</p>  <p>Potentiometer</p> 
	<p>Read a digital input (pin 2 or pin 3)... stores a number 0 or 1.</p> 	<p>Switch</p> 
<u>OUTPUTS</u>		
 	<p>Turn ON or Off a digital pin (#10 – 13)</p>	<p>LEDs</p>  <p>Transistor (later)</p>

			
analog 3 value 255	<p>Write a Pulse Width Modulation (PWM) signal to pin 5, 6, or 9.</p> 	<p>LEDs</p> 	<p>Relay (optional)</p> 
motor 7 angle 180	<p>Rotates a servomotor to a specific position, pins 4, 7, 8</p> 	<p>Servomotor</p> 	

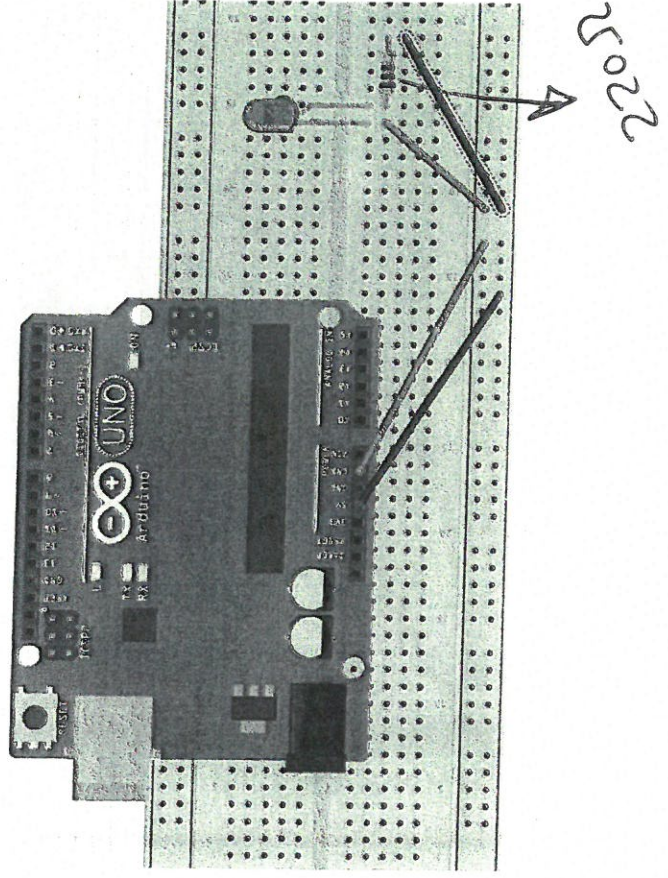
Basic Arduino Commands

<code>void setup() { }</code>	Code between the curly brackets is run once whenever the Arduino is turned on or reset
<code>void loop() { }</code>	Code between the curly brackets is run over and over again after the setup function
<code>delay(500);</code>	Arduino goes to "sleep" for 500 ms (1/2 second)
<code>pinMode(2, OUTPUT);</code>	Pin 2 on the Arduino is set to be an OUTPUT; it can turn on or off something
<code>pinMode(2, INPUT);</code>	Pin 2 on the Arduino is set to be an INPUT; the state of the pin can be read as either HIGH or LOW
<code>pinMode(2, INPUT_PULLUP);</code>	Similar to above, but pin 2 is pulled HIGH by default. It can be set LOW by connecting it to ground (GND).
<code>digitalWrite(13, HIGH);</code>	Set Pin 13 to output 5 Volts.
<code>digitalWrite(13, LOW);</code>	Set Pin 13 to output 0 Volts (GND).
<code>digitalRead(2)</code>	Stores the value of the voltage on Pin 2: A voltage close to 0 V (GND) registers as a "0". A voltage close to 5 V registers as a "1".
<code>Serial.begin(9600);</code>	Begin a serial connection to your computer at a rate of 9600 bits per second.
<code>Serial.print("Hello"); Serial.print(x);</code>	Display "Hello" or the value of variable x on the computer screen.
<code>Serial.println("Hello"); Serial.println(x);</code>	As above, but also move to a new line (like an "Enter" key)
<code>if(x > 5){ }</code>	Do the commands between the curly brackets if a variable x has a value greater than 5.
<code>if(x > 5){ } else { }</code>	Do the commands between the FIRST curly brackets if a variable x has a value greater than 5. Otherwise do the commands between the SECOND curly brackets.

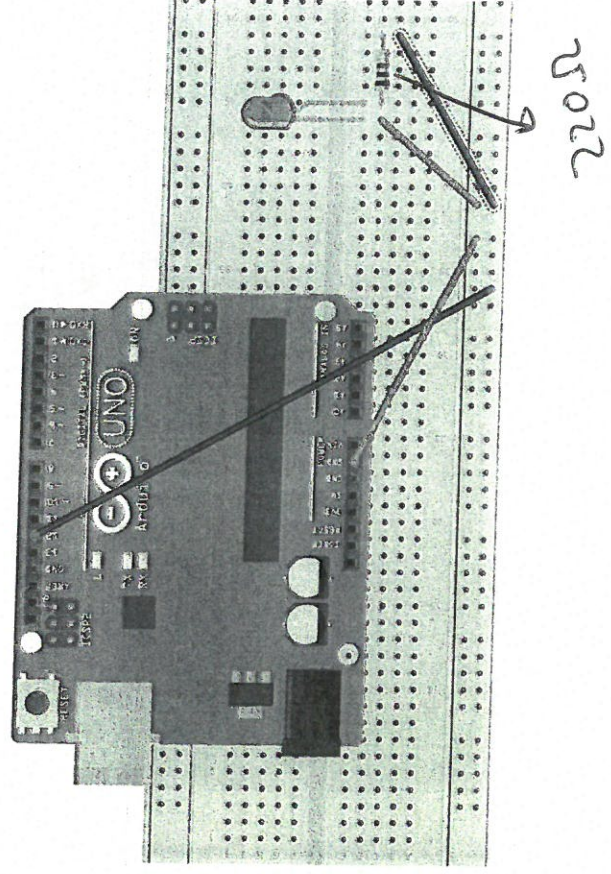
<pre>if(num == 1){ } else if (num == 2) { } else { }</pre>	<p>Do the commands between the FIRST curly brackets if a variable num equals one. Do the commands between the SECOND curly brackets if a variable num equals two. Otherwise do the commands between the THIRD curly brackets.</p>
<pre>for(int i = 0; i < 10; i++) { }</pre>	<p>Do the commands in curly brackets 10 times, while i has the values 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each time through the loop, the value of i increases by one.</p>
<pre>int x = 5;</pre>	<p>Define a 16 bit variable called x and give it a value of 5.</p>
<pre>long x = 1000000;</pre>	<p>Define a 32 bit variable called x and give it a value of one million.</p>
<pre>float y = 0.5;</pre>	<p>Define a 32 bit variable called y and give it a value of one half.</p>
<pre>analogRead(A2)</pre>	<p>Holds the value of analog pin2, as a number between 0 and 1023 (10 bit integer)</p>
<pre>analogWrite(9, 128)</pre>	<p>Pulse digital pin 9 with Pulse Width Modulation (PWM). The second number can be between 0 and 255 (8 bit). This can only be done with PWM digital pins (indicated with a ~ sign on the Arduino).</p>
<pre>#include <Servo.h></pre>	<p>Include commands necessary to use Servo objects</p>
<pre>Servo myservo;</pre>	<p>Create an instance of the Servo class called myservo. You can name it anything you want to.</p>
<pre>myservo.attach(3);</pre>	<p>The servo has its control wire (usually yellow colored) attached to pin 3 on the Arduino. This command is placed in the setup function.</p>
<pre>myservo.write(90);</pre>	<p>Tell the servo to go to position 90°. Typically, servo operate over 180 degrees of position.</p>

LED Tasks

#1: Use the Arduino to power an LED

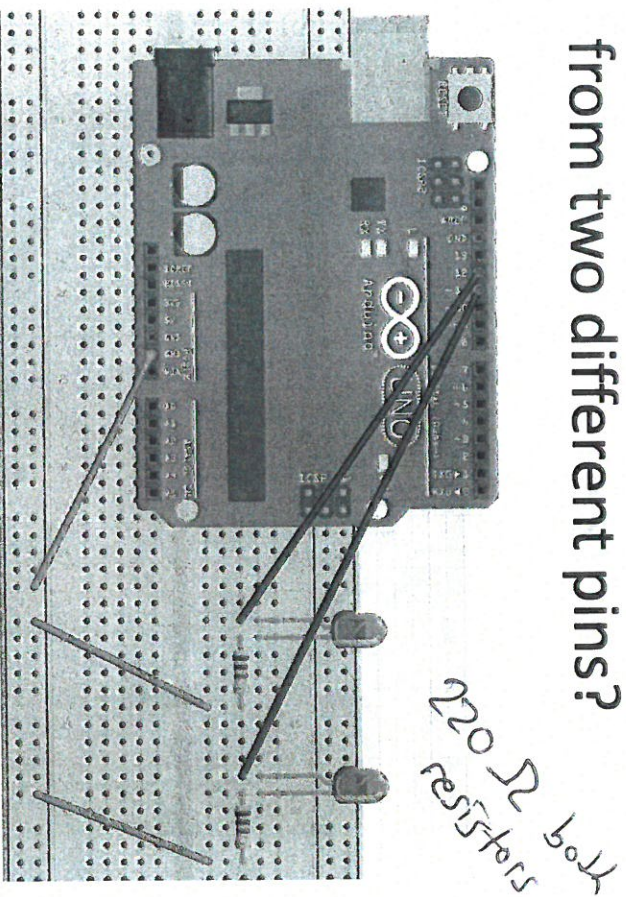


#2: Use pin 12 to power an LED



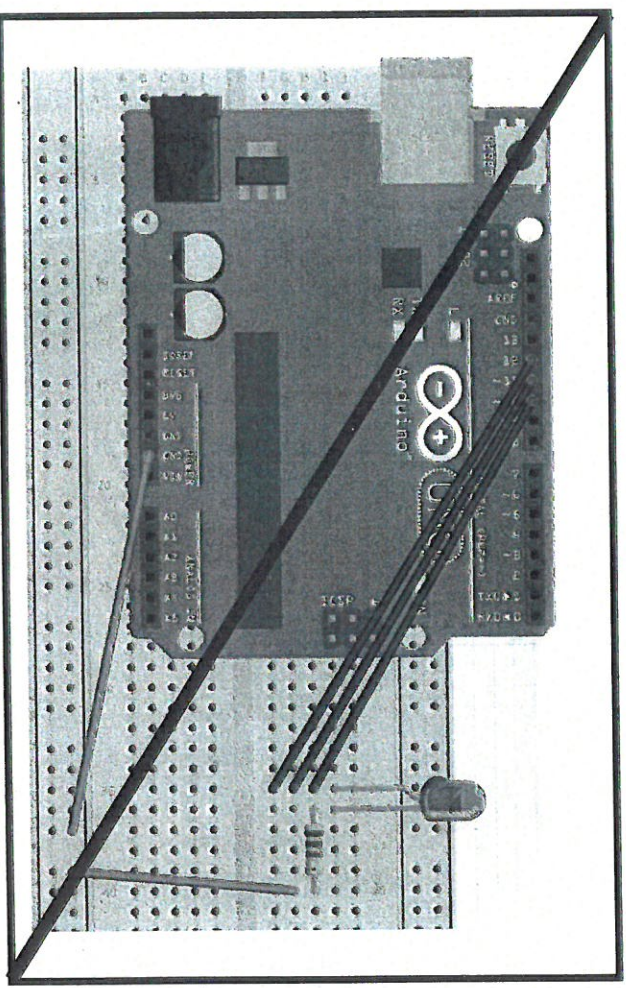
LED Tasks

#3: How can you turn on two different LEDs from two different pins?

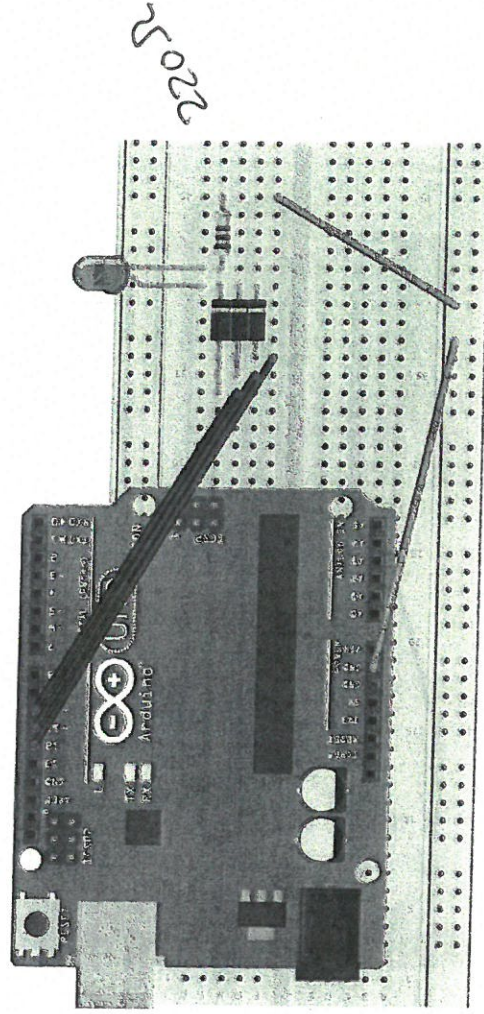


#4: How can you turn on an LED when any of three pins is turned on?

This doesn't work... why not?



LED Tasks



Scratch 4 Arduinos Tasks

Analogous Tasks in Arduino IDE

Arduino Programming

- Chapter 1 - Course Introduction_The History of a Robot and the Story of Arduino.pdf 2 MB
- Scratch for Arduino
 - Intro to S4A.pdf 7 MB
 - Sample S4A Project
- Chapter 3 - XYZrobot SCRATCH.pdf 2 MB
- Resources to Read
- Assignments to Do
- Extras: Going Further
- Basic Arduino Commands.docx 14 KB

Tasks

"Answers"

Arduino Programming

- Chapter 1 - Course Introduction_The History of a Robot and the Story of Arduino.pdf 2 MB
- Scratch for Arduino
 - Intro to S4A.pdf 7 MB
 - Sample S4A Project
- Chapter 3 - XYZrobot SCRATCH.pdf 2 MB
- Resources to Read
- Assignments to Do
 - Chapter 4 - The basic control of LED and applications of resistor and switch.pdf 1 MB
 - Using Switches to Control LEDs.docx 397 KB
 - Questions on "Using Switches to Control LEDs" docx 42 KB
 - Write the Outputs for each of the Loops.docx 113 KB
 - Write the Outputs for each Sketch which uses Variables.docx 12 KB
 - Input_potentiometer.pdf 175 KB
- Extras: Going Further
- Basic Arduino Commands.docx 14 KB

Same as Task #2 in S4A

pg 9-12

Similar to Task #3 in S4A

Same as Task #1 in S4A

Week 5 Essay and Reflection
TUESDAY 1/17 12:35 pm

Week 6 Essay and Reflection
MONDAY 1/23 12:35 pm

Week 7 Essay and Reflection
MONDAY 1/30 12:35 pm

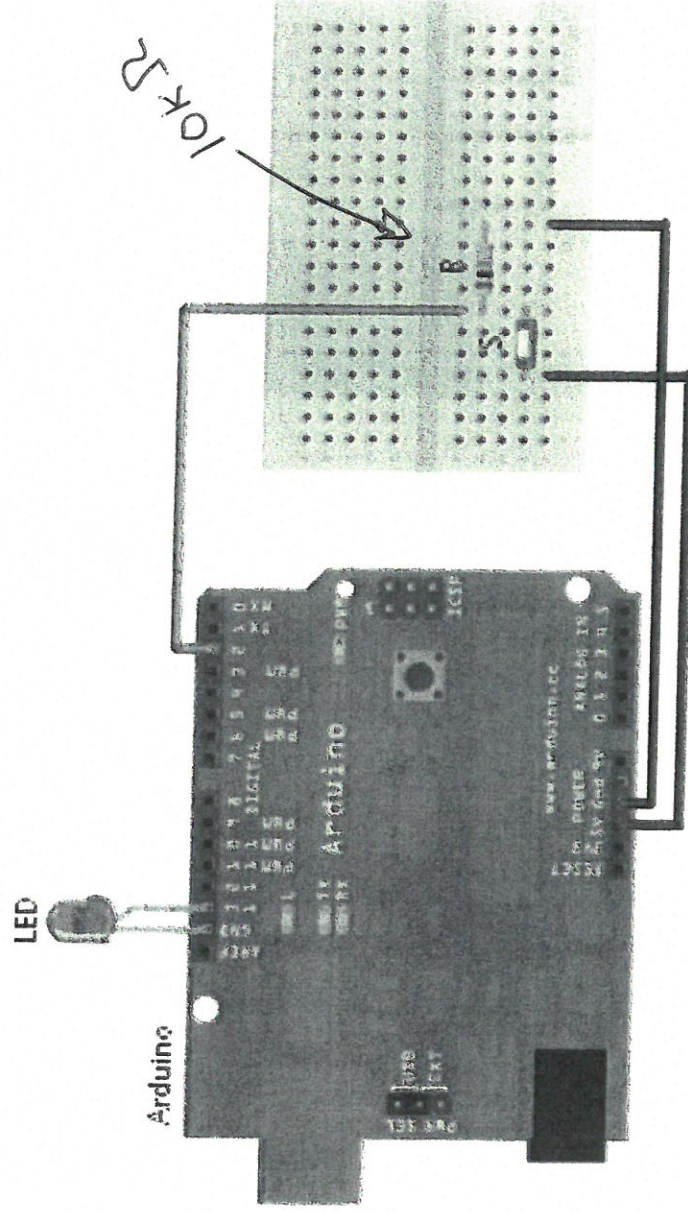
Week 8 Essay and Reflection
MONDAY 2/6 12:35 pm

Week 9 Essay and Reflection
MONDAY 2/13 12:35 pm

Week 10 Essay and Reflection
MONDAY 2/20 12:35 pm

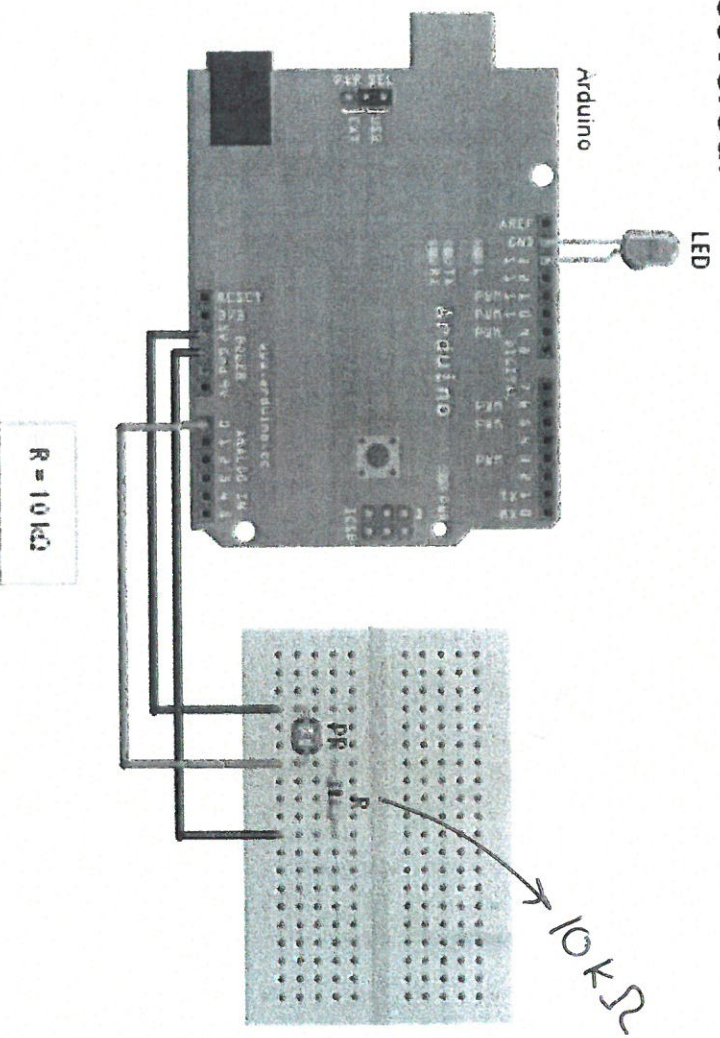
S4A Task #1

- Use a switch to turn on an LED.



S4A Task #2

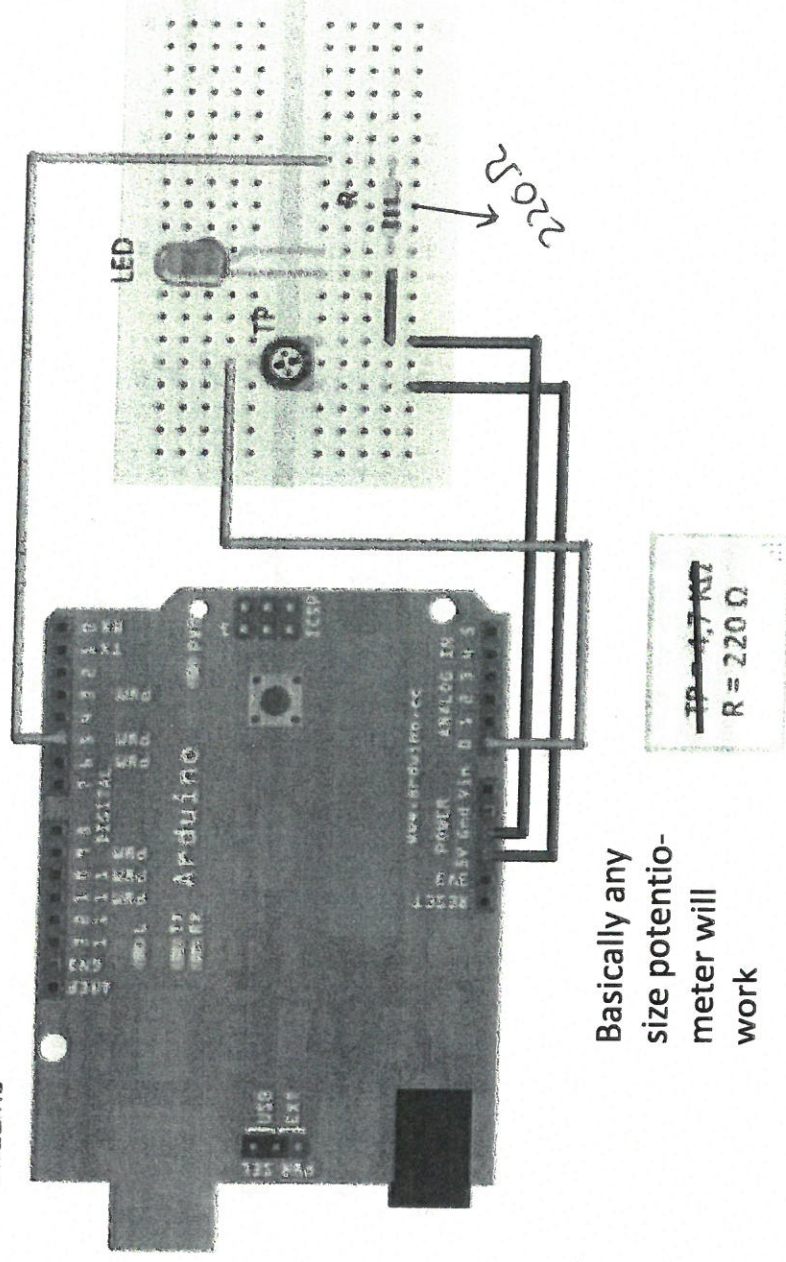
Now we will replace the button by a LDR (Light Dependent Resistor). The objective is turn on the LED when the LDR is covered.



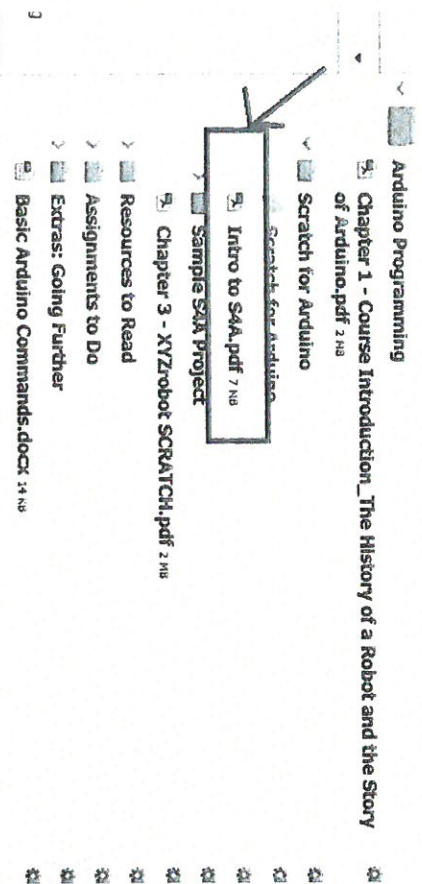
Now we're going to use a potentiometer instead of fotoresistor, with which we'll control the light of the LED gradually. This means that we need to use a PWM pin (for example 5) instead of the pin 13:

Arduino

Task #3



Additional Exercises (see here Schoology)



Ex. 4: Semaphore

In this exercise we'll try to simulate the operation of a traffic light, those which have a button to facilitate the passage of pedestrians. The goal is to simulate the behavior of these lights with a simple program.

Ex. 5: RGB LED + Infrared sensor

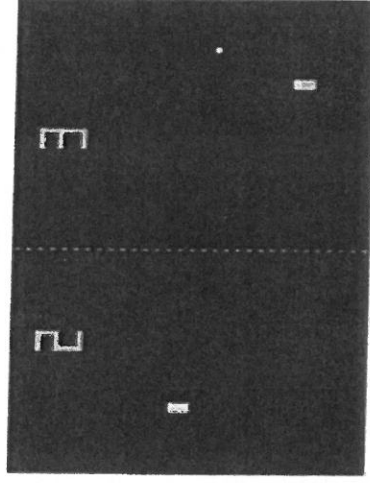
The goal of this exercise will be the interaction between objects, using messages or variable changes. We'll also see and use the lists for the first time, and the elements in programming as **flags**.

To do so entertaining we will create a "virtual musical instrument."

This will be the first time we program more than one object. The goal is to split the program into different objects and that each has a different function.

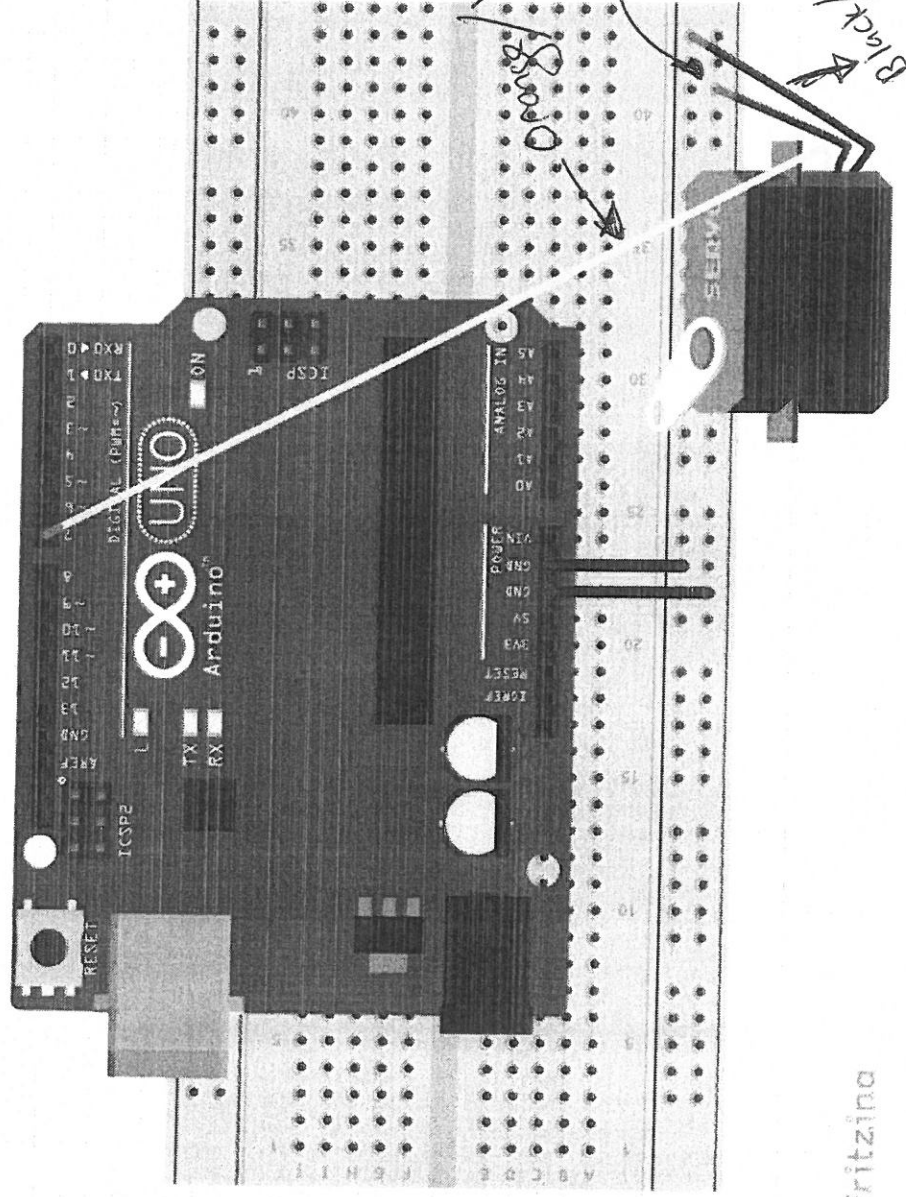
Ex. 6: Pong with sensors

Now we'll try to remake one of the first video game in history, the PONG:

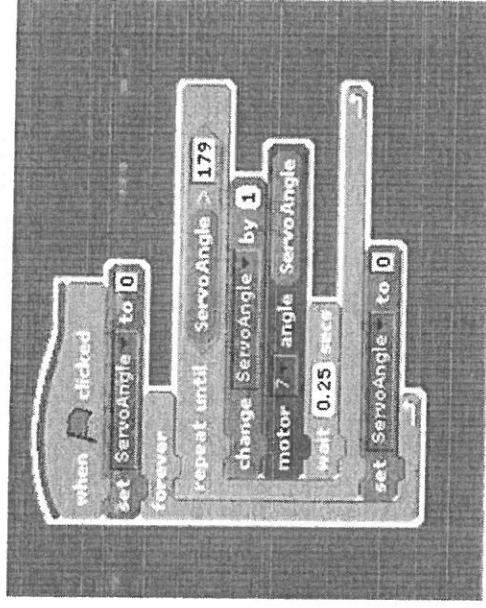


The goal is to use two sensors on the same board, and assign one to each player. In addition, we can incorporate a button to start the game, LEDs that light every time the players score a point, and so on...

Servo Motor



fritzing



Rotates to a given position and holds that position. Position can be any number between 0 and 180.

