Jacobs University Bremen

Introduction to Robotics and Intelligent Systems Lab

(Spring 2020)

Lab 2

Group Members :

- Hamza Bouhelal

- Mohamed Reda Arsalane

-Badr Essefiany

-Salah Eddine Naouassih

1-Introduction:

In this lab we are getting in touch with the material and Arduino, working with different codes and equipment to solve the tasks we are given. This report is mainly constituted of pictures videos and explanations of what we did, the problems we faced, and how we solved them.

2-Lab Tasks:

Task 2.1:

In this task, we measured the resistance of the LDR twice, once covered with a paper, once without it.

-Value of the resistance with the paper: 7.6* 200k ohm = 1520kΩ

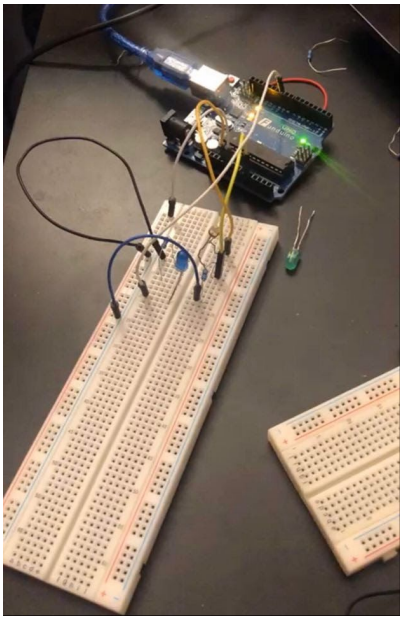
-Value of the resistance without the paper: 2.9* 200k ohm = 580kΩ

With the lights on and a paper in between.

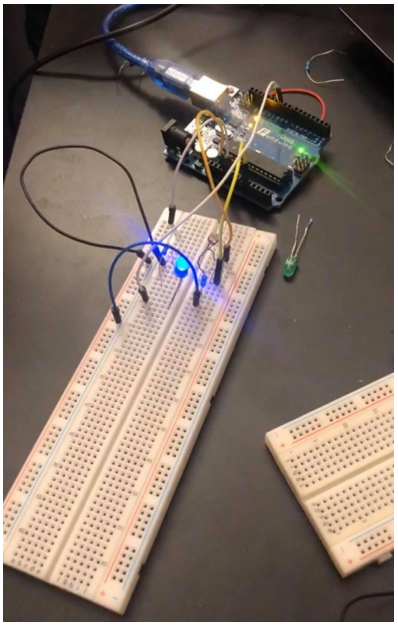
same experience with no cover

Task 2.2:

We built the system with a resistance of 1K ohm and noticed that the LED didn't light up, when we switched the resistance to a 10k ohm resistance the Led turned on.



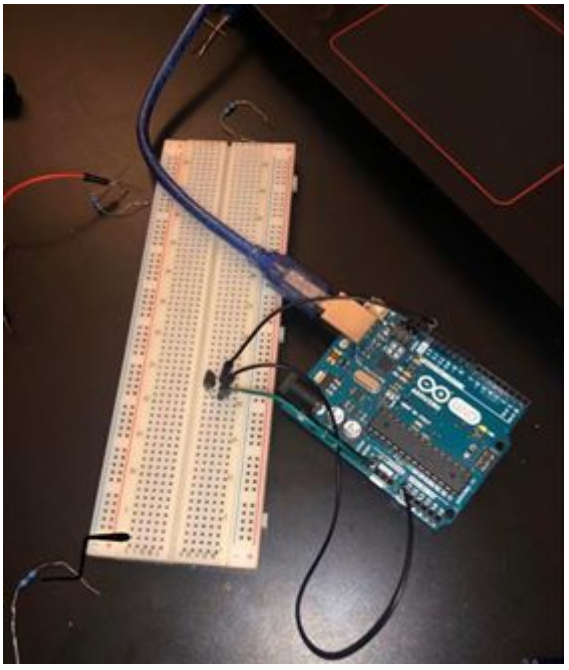
The light off

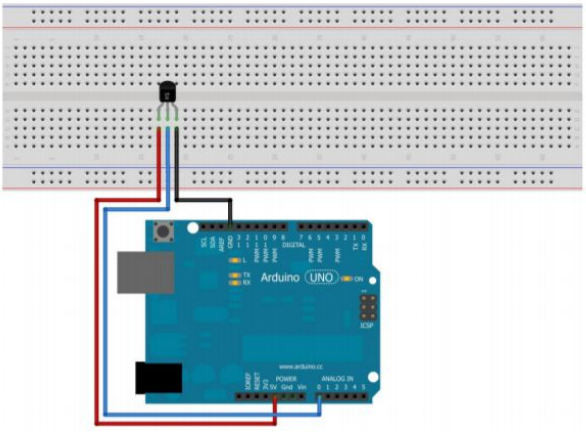


The light on

The video:
https://www.youtube.com/watch?v=fT0fbOKSzFM

Task 2.3:





We set up the circuit as shown in the figure, and touch the sensor to change its temperature. We can see afterwards the fluctuation when opening the serial monitor shown below

Task 2.4:

The temperature sensor outputs the value of the temperature in the form of a voltage between 0V and 2V, using the analogRead (TMP36) command this voltage is recognized as a value between 0 and 410. Thus 410 is the maximum numerical value read by the analogRead (TMP36) command equivalent to a 2V voltage.

Then, the map command converts the numerical value (between 0 and 410) into a temperature value in C° (between - 50 and 150).

Task 2.5:

```
const int trig = 7;
const int echo = 6;
const int buzz =4;
void setup ()
{
pinMode ( trig , OUTPUT ) ;
pinMode ( echo , INPUT ) ;
pinMode ( buzz , OUTPUT ) ;
Serial . begin (9600) ;
}
void loop ()
{
digitalWrite ( trig , HIGH ) ;
delayMicroseconds (15) ;
digitalWrite ( trig , LOW) ;
long duration = pulseIn ( echo , HIGH ) ;
const long vsound = 340;
long dist = ( duration / 2L ) * vsound / 10000L ;

if ( dist < 100L && dist > 20L )
{
```

```
digitalWrite ( buzz ,HIGH );
delay(dist*50);
digitalWrite ( buzz ,LOW );
delay(dist*50);
Serial . println (dist) ;
}
else if ( dist < 20L )
{
digitalWrite ( buzz ,HIGH );
Serial . println (dist) ;
}

else
{
Serial . print ( dist ) ;
Serial . println (" cm") ;
digitalWrite ( buzz , LOW);
}

}
```

Task 2.6:

Pins 9 and 10 are the first Uno pins I use for servos because using the Servo.h library disables the analogWrite() command to pins 9 and 10 anyway.

Task 2.7:

```
# include <Servo .h>
Servo s ; // create servo object;
void setup ()
{
s . attach (8) ; // control signal on pin 8
```

```
}
void loop ()
{

s . write (0) ;
delay (3000) ;
s . write (45) ;
delay (3000) ;
s . write (90) ;
delay (3000) ;
s . write (135) ;
delay (3000) ;
s . write (0) ;
delay (3000) ;
}
```

We didn't find an unsymmetrical white plastic flanges so we did it with a regular one.

<u>Task 2.8:</u>

```
#include <Servo.h>
Servo s ; // create servo object ;
void setup ()
{
s . attach (8) ; // control signal on pin 8
}
void loop ()
{
// now we can control the servo with write ( angle )
// angle of the servo 0 -180 degrees
// 90 degrees = servo is centerd
s . write (0) ;
delay (3000) ;
s . write (180) ;
```

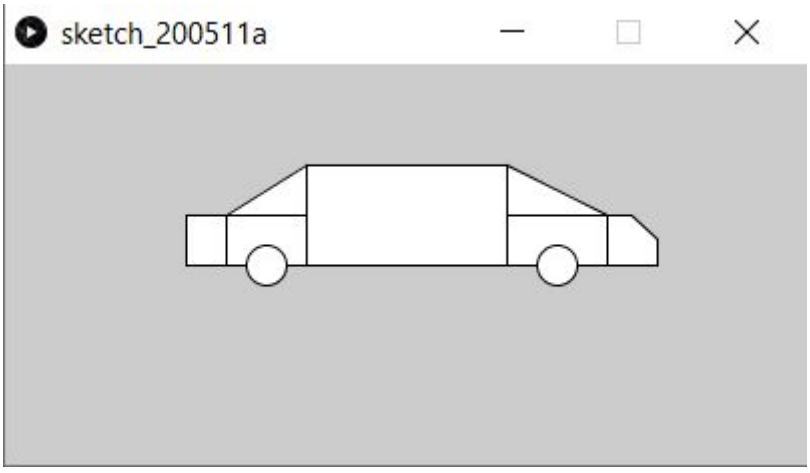
delay (3000) ;

The video:
https://www.youtube.com/watch?v=KJOtJq9UER0

Task 2.9:

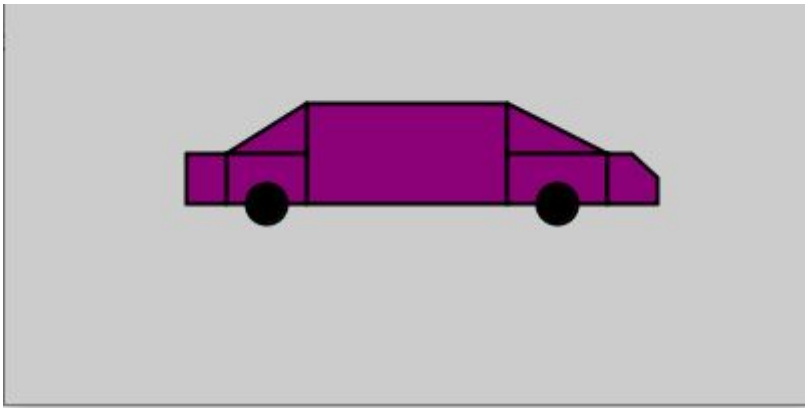
Using the codes provided in the lab manual, we can try to build a car looking shape, mainly using the rectangle codes, sphere and triangle ones. We also implemented the vertex codes to draw any shapes we'd want, in this particular situations: irregular pentagons

```
size(400,200);
rect(150,50 ,100 ,50);
triangle(250,50 ,250 ,75 , 300, 75);
rect(250,75 ,50 ,25);
triangle(150,50 ,150 ,75,110 ,75 );
rect(110,75,40,25);
beginShape();
vertex( 312, 75);
vertex( 325,87 );
vertex( 325,100 );
vertex(300 ,100 );
vertex(300 ,75 );
endShape(CLOSE);
rect(70,75 ,40 ,25);
ellipse(275,100 , 20, 20);
ellipse(130,100 , 20, 20);
```



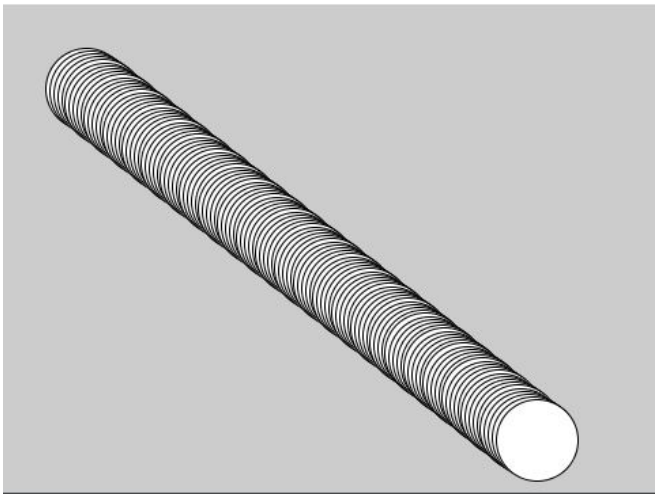
Task 2.10:

```
size(400,200);
strokeWeight(2);
strokeJoin(ROUND);
fill(140,0,120);
rect(150,50 ,100 ,50);
triangle(250,50 ,250 ,75 , 300, 75);
rect(250,75 ,50 ,25);
triangle(150,50 ,150 ,75,110 ,75 );
rect(110,75,40,25);
beginShape();
vertex( 312, 75);
vertex( 325,87 );
vertex( 325,100 );
vertex(300 ,100 );
vertex(300 ,75 );
endShape(CLOSE);
rect(90,75 ,20 ,25);
fill(0,0,0);
ellipse(275,100 , 20, 20);
ellipse(130,100 , 20, 20);
```



Task 2.11:

When removing *background (102) ;* from the code, we can observe the following result, where the circles pile up on each other.



This is due to the background not being drawn again each time we go in into the draw loop, which repeats itself continuously; When moving *background (102) ;* in setup() we only draw the background once, which causes the circles to be drawn over the same previous drawing.

Task 2.12:

The X-Coordinate of the mouse motion is controlling the brightness of the LED.