****

ARCH-581DL: Interactive Design  
Assignment Report – Authors: Helia Zakeri / Andrea Menardo

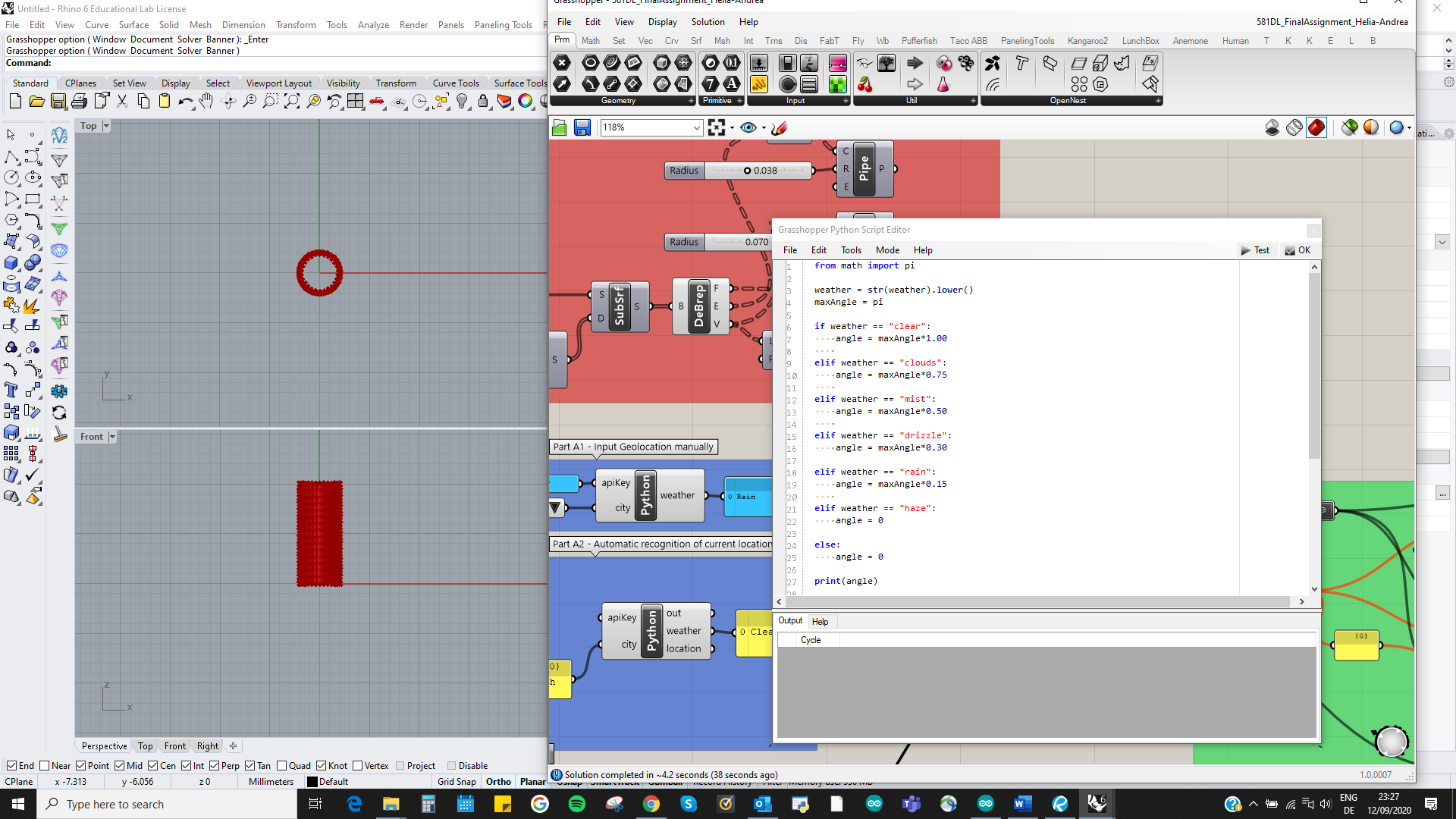
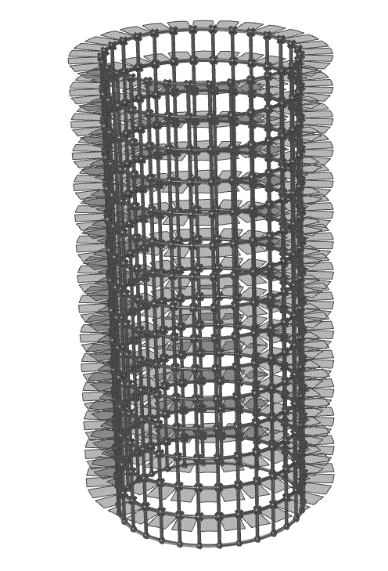
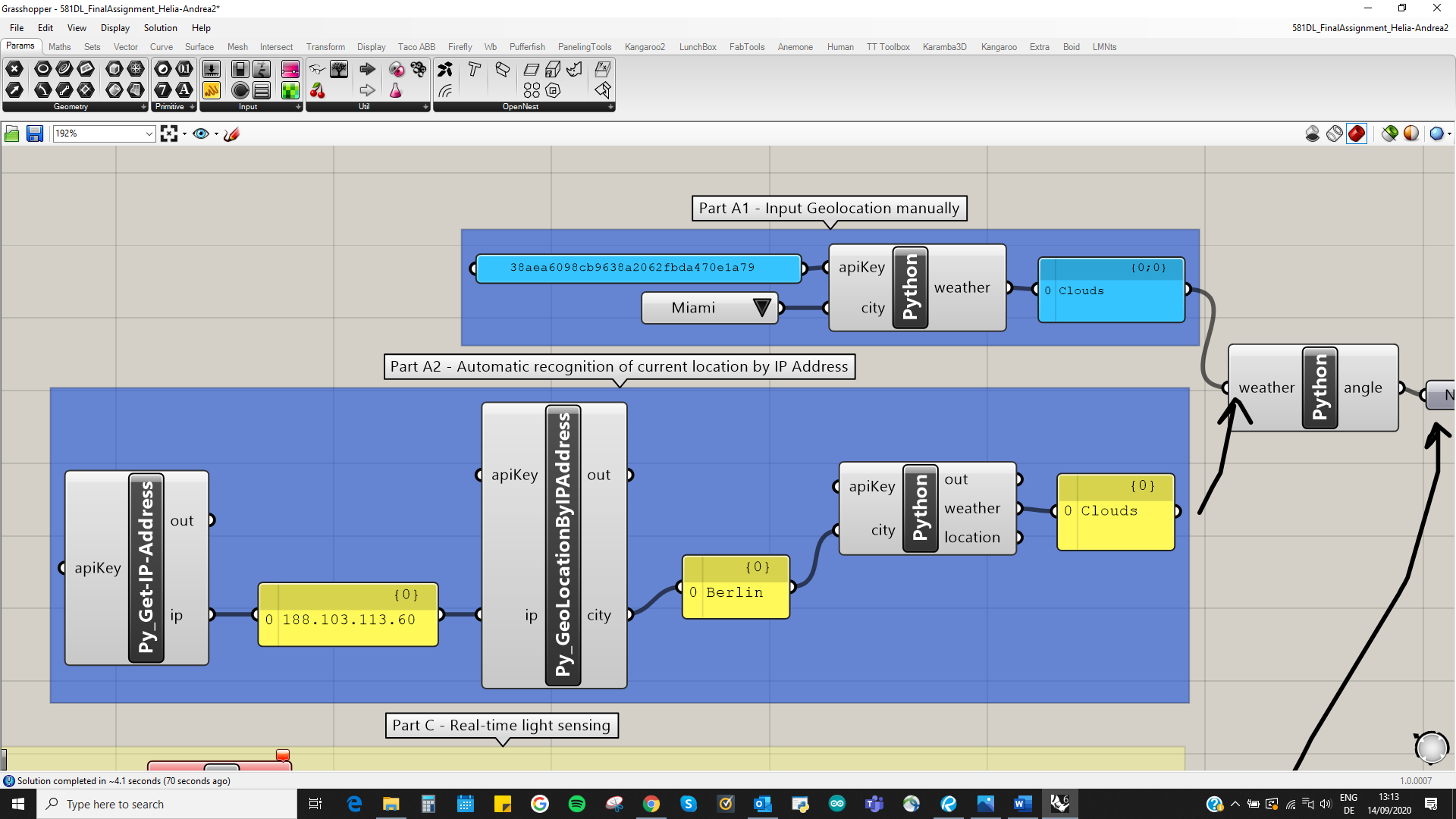
*[ARC] Department of Architecture, University of Nicosia, Nicosia, Cyprus*

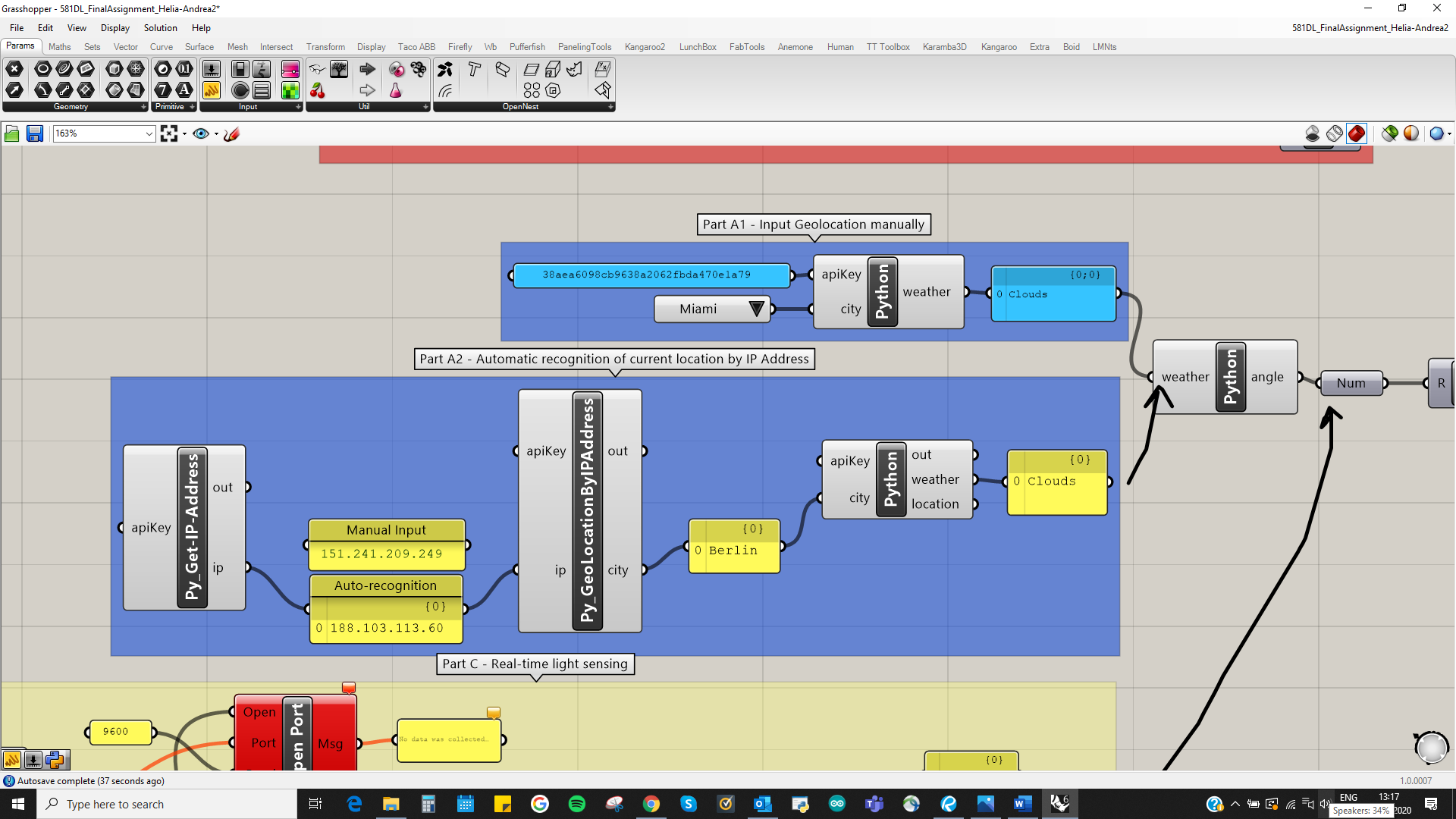
*MSc in Computational Design and Digital Fabrication Spring semester 2020*

**Interactive Windows**

Part A: City-based

The following GH definition defines and changes the rotation angle of the windows. When set to city-based, we used an API key from <http://openweathermap.org>, outputting the weather in a specific city. The weather variable is then converted into an angle in degrees using a python script, where a series of “if” functions assign a different angle to a different weather condition: for example, when the weather is clear, windows will be fully open and the angle 180° or, if it is rainy, the windows are almost closed meaning the angle is 0°. in case the maximum angle a window can have is 180 degrees when fully open, we set the angle for different types of the weather.



Part B: Automatic Location

Alternatively an API key from <https://ipstack.com/> outputs the user’s current location by tracing the IP address. To get the IP address automatically we used <https://api.ipify.org> and a separate script. This allows to override the string in the panel, in case a specific IP needs to be types.

Part C: Real-time Light Sensing

For this task the Arduino Nano available to Helia was used, with a phototransistor connected to the GH definition through the Firefly plugin. This enabled reading values from 0 to 1023, which were remapped in the domain of 0.0 to 1.0, representing the amount of sunlight the building gets. To convert sunlight intensity into angle variables the sunlight value was multiplied in π to get values in radians and fed to the Python definition to make changes on the windows rotation in real-time.

const int PHOTO\_PIN = A1;

const int LED\_PIN = 13;

int sensorValue;

int sensorLow = 1023;

int sensorHigh = 0;

void setup() {

pinMode(LED\_PIN, OUTPUT);

digitalWrite(LED\_PIN, HIGH);

while(millis() < 5000) {

sensorValue = analogRead(PHOTO\_PIN);

if(sensorValue > sensorHigh) {

sensorHigh = sensorValue;

}

if(sensorValue < sensorLow) {

sensorLow = sensorValue;

}

}

digitalWrite(LED\_PIN, LOW);

Serial.begin(9600);

}

void loop() {

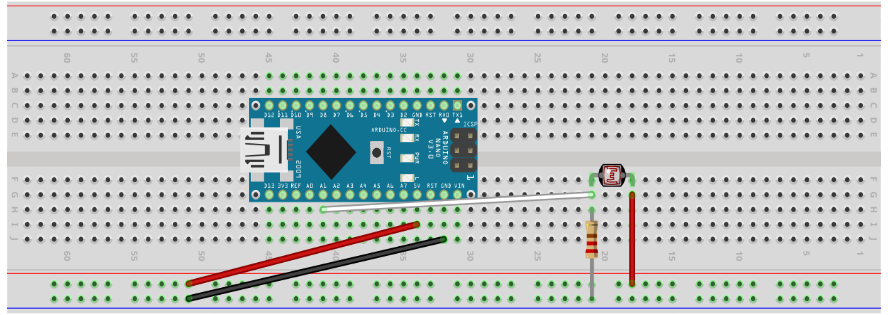
// put your main code here, to run repeatedly:

sensorValue = analogRead(PHOTO\_PIN);

Serial.println(map(sensorValue, sensorLow, sensorHigh, 0, 1023));

delay(5000);

}



Part D: Window Servo Control

For this part the Arduino UNO available to Andrea was used. The sketch Firefly\_Firmata is first uploaded to the Arduino board to open a connection between GH and the board. Since the servo shows angles in degrees from 0 to 180°, we converted the outcome angles of parts A, B or C from radians to degrees before feeding to pin ~9 (input set on “servo”), where the servo motor was connected to. A video of a rough mock-up of the building shows the motor changing the angle of the windows in real-time.

#include <Servo.h>

Servo motor;

const int motor\_pin = 9;

int val\_angle;

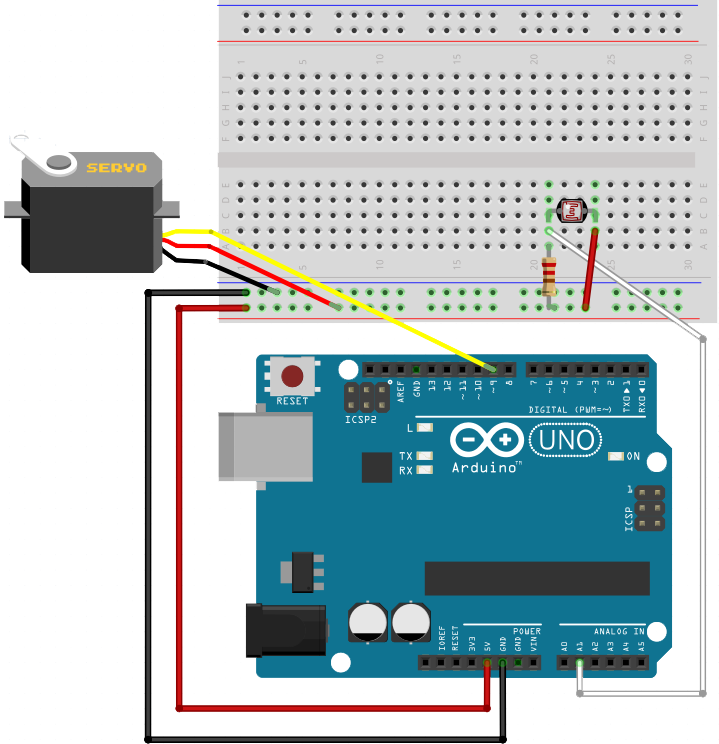
void setup() {

motor.attach(9);

}

void loop() {

}



From part C