Gesture Control Robocar

**A Project Report submitted in partial fulfillment of requirement for the award of degree**

**Bachelor of Engineering**

**In**

**Electronics &Communication Engineering**

**Submitted by:-**

**Shashank Jain** (1910992001) **Project Guide:**

**Ayush Sharma** (1910992009) Mr. Amit Pandey



**May 2021**

**Department of Electronics &Communication Engineering**

**Chitkara University, Punjab**

##### **CERTIFICATE**

We “Shashank Jain (1910992001) and Ayush Sharma (1910992009)” hereby declare that this project report entitled **“Gesture control Robo-car”** is an authentic record of our own work carried out towards the Major Project, done at Chitkara University Institute of Engineering & Technology, Rajpura, Punjab.

The project has been carried out under the guidance of “**Mr. Amit Pandey”** during 4th semester (February, 2021 to May, 2021).

**Signatures:**

|  |  |
| --- | --- |
| **Shashank Jain** |  |
| **Ayush Sharma** |  |

This is certified that above statement made by the students is correct to the best of my knowledge and belief.

**Guide Name (Dr. Shivani Malhotra)**

**(Project Guide) ( Dean, ECE)**

**ACKNOWLEDGEMENT**

We would like to thank our teacher Amit Sir as our project advisor for his continuous effort in helping us develop our project from the initial stages till completion. His support and expertise have been an essential driving force for us and has an immense share in the success of our project.

Priyanka Mam is a person who has also played an essential role in this endeavor. She guided us throughout the step-wise growth of the project. We would also like thank her for providing us the technical support in every division of this project. Without her support, this project would have been quite a difficult task to achieve. We are very thankful for all his time and patience with us.

We would also thank all our friends who have contributed towards the completion of this project. We will always be indebted to them.

The research of this project has helped me to gain more knowledge and gave me direction how I can convert my ideas to reality.

**CONTENTS**

1. **INTRODUCTION Page No.**
2. About the project 5
   1. Literature review 5
3. **PROJECT PLANNING**
4. Block Diagram & Description 7
   1. Specifications 8
5. **PROJECT DETAILS**
6. Major Components of Project 8

3.2 Working of the project 13

3.2 Images of Final Project/Output waveforms 13

1. **SOFTWARE DEVELOPMENT** 17
2. **RESULTS AND CONCLUSION** 23

**6. REFERENCES** 24

CHAPTER 1

Introduction

* 1. **About the project:**

A gesture-controlled robocar is controlled by using a hand in place of any other method like buttons or joystick.[1] Controlling a robocar using a gesture is a better, easy and more effective method in comparison to using a remote control.[2] Here by giving different gestures you can control the car.

This car can be used in many applications. It all depends upon the user as it is on your imagination how you can use it. As you have a robot whom you can control using hand gestures. Maybe by adding one more component in this gesture control car you can make more effective use of this car. There are some problems also which can occur in this project that maybe sometimes your gesture controller was not able to identify the user gesture because there are so many gestures that a user can make and maybe your robocar is not moving with the desired speed. So as a user I can make many gestures from hand, so we used a gesture controller that can recognize 9 gestures. The components that we are going to use in our project are all very efficient.

* 1. **Literature Review**

Humans in the physical world interact by the means of the five senses. Gestures play important role in the field of communication from ancient time, even before the invention of any language. These days machines are taking control over every complex work, interactions with machines have become really important.[1]

Robotic technology is growing every day and it is helping us in various tasks. Successful applications of large multirobot teams include industrial automation [2]

Typical modalities used to remotely control robots, such as keyboard, mouse, joystick or teach pendant, are generally considered unnatural methods of communication and require prior training, which can be unpleasant and time consuming. There are many ways of controlling a robocar, such as keyboards, mouse, joysticks or teach pendant but these methods are unnatural ways of communication which requires a prior training, which can be unpleasant and time consuming. Here we will be going to use gesture control sensor to control our robocar. [4]

Gesture sensors can sense a lot of different types of human gestures. The gesture sensor can capture human gestures using a camera, or a data glove. There are many waves to capture the gestures like via Bluetooth or infrared waves, Acoustic, Tactile, optical or motion technological means.[1]

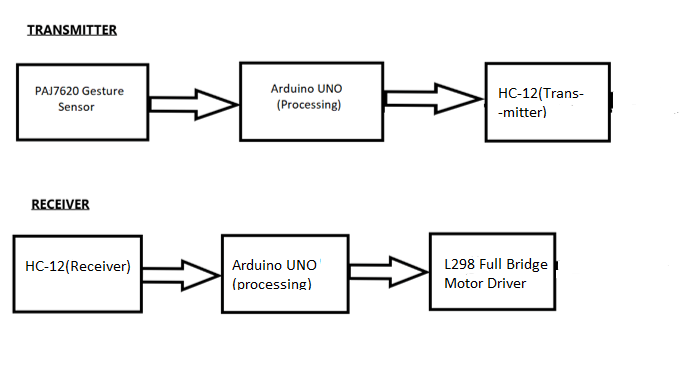
Gesture Control robocar is a kind of robot which can be controlled by our hand gestures not by old buttons. This gesture control robocar can be used in many ways and we believe that its applications will only going to increase with time. As it is not something which have predefined finite number of applications. It depends on user who can think of more creative ways to use it.

This paper focuses on creating a robocar with the help of gesture sensor, wireless module, motor driver and a microcontroller.[3] The central goal of the paper is to implement a system through which the user can give commands to wireless Robot using gesture.

CHAPTER-2

Project Planning

**2.1 Block Diagram**



**CHAPTER 3**

**Project Details**

**3.1 Major Components of Project**

**3.1.1 Arduino UNO:**

The **Arduino Uno** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits. The board has 14 digital I/O pins (six capable of [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) output), 6 analog I/O pins, and is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment), via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

* [Microcontroller](https://en.wikipedia.org/wiki/Microcontroller): [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) [[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)
* Operating Voltage: 5 Volts
* Input Voltage: 7 to 20 Volts
* Digital I/O Pins: 14 (of which 6 can provide PWM output)
* UART: 1
* I2C: 1
* SPPI: 1
* Analog Input Pins: 6
* DC Current per I/O Pin: 20 mA
* DC Current for 3.3V Pin: 50 mA
* [Flash Memory](https://en.wikipedia.org/wiki/Flash_Memory): 32 KB of which 0.5 KB used by [bootloader](https://en.wikipedia.org/wiki/Booting#BOOT-LOADER)
* [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory): 2 KB
* [EEPROM](https://en.wikipedia.org/wiki/EEPROM): 1 KB
* Clock Speed: 16 MHz
* Length: 68.6 mm
* Width: 53.4 mm
* Weight: 25 g

**3.1.2. L298 Full motor bridge driver:**

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

* OPERATING SUPPLY VOLTAGE UP TO 46V
* LOW SATURATION VOLTAGE
* TOTAL DC CURRENT UP TO 4A
* LOGICAL \"0\" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
* OVERTEMPERATURE PROTECTION

**3.1.3. HC-12 Wireless Module:**

The HC-12 is a half-duplex wireless serial communication module with 100 channels in the 433.4-473.0 MHz range that is capable of transmitting up to 1 km. This project will begin by using the HC-12 to create a wireless link between two computers and end with a second article that creates a simple wireless GPS tracker.

 Frequency range: 119–1050 MHz

 Receive sensitivity: –126 dBm

 Serial TTL (RX, TX, GND) interface

 Modulation (G)FSK, 4(G)FSK, (G)MSK, OOK

 Max output power: +20 dBm

 Low active power consumption: 10/13 mA RX, 18 mA TX at +10 dBm

 Ultralow current power down modes: 30nA shutdown, 50nA standby

 Data rate: 100 bps to 1 Mbps

 Fast wake and hop times

 Highly configurable packet handler

 TX and RX 64byte FIFOs

 Auto frequency control (AFC)

 Automatic gain control (AGC)

 Low BOM

 Low battery detector

 FCC Part 90 Mask D, FCC part 15.247, 15,231, 15,249, ARIB T-108, T-96, T-67, RCR STD-30, China regulatory

 ETSI Class-I Operation with SAW

**3.1.4. PAJ7620(gesture sensor):**

PAJ7620U2 integrates gesture recognition function with general I2C interface into a single chip. It can recognize 9 gestures and the gesture information can simply access via the I2C bus. PAJ7620U2 also offers built-in proximity detection for the purpose of sensing object approaching or departing. The PAJ7620U2 is designed with great flexibility in a power-saving mechanism. The PAJ7620U2 is designed to operate from 2.8V to 3.3V over -40°C to +85°C and the pull-up voltage for the I2C bus and interrupt line is from 1.8V to 3.3V.

1. The supply voltage is 2.8V- 3.3V and I/O voltage is 1.8V~3.3V
2. Nine gesture recognition (Up / Down / Left / Right / Push / Pull / CW / CCW / Wave)
3. Gesture speed is 60°/s to 600°/s in Normal Mode and 60°/s to 1200°/s in Gaming Mode
4. Ambient light immunity: < 100k Lux
5. Built-in proximity detection
6. Flexible power-saving scheme
7. I2C interface up to 400 kbit/s, Pull-up voltage from 1.8V to 3.3V
8. Ambient light noise cancellation

**3.2 Working of the project:**

We will give hand gestures and our robocar will be going to move according to it. We will use a tiny little sensor that can recognize various hand gestures such as moving your hands up, down, left, right, forward, backward, clockwise, anti-clockwise, and waving.

Here we will be using the HC12 wireless module for sending data from the Sensor to the robocar. It is one of the most commonly used wireless modules in the field of robotics and other remote-control applications.

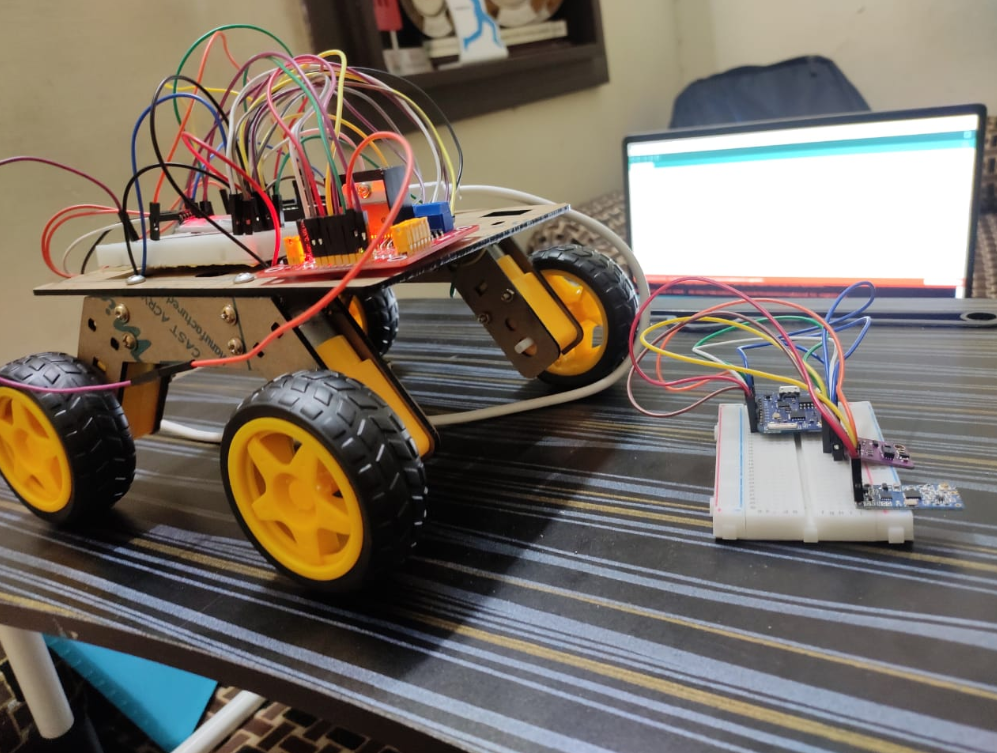
Then we have the transmitter of the Gesture Controlled Robocar consist of a Gesture sensor that senses the gesture, an Arduino that processes the signal, and an HC12 Wireless module that will transmit the signal to the Receiver.

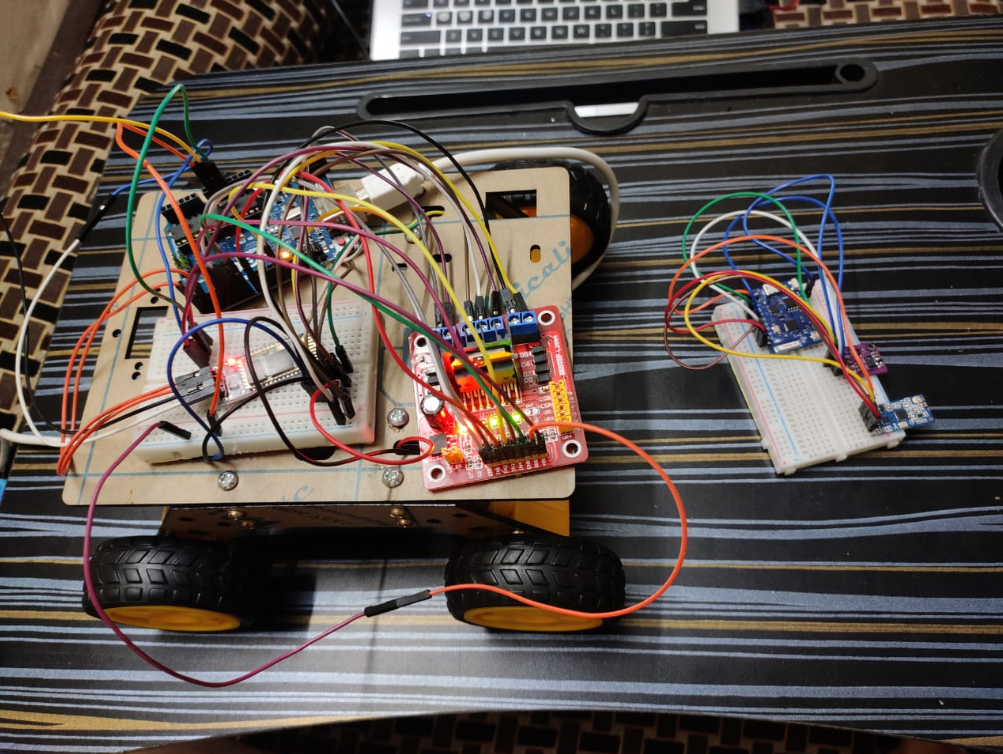
The Receiver part consists of an HC12 wireless module that receives the signal from the Transmitter module, an Arduino which will process the signal and drive our Arduino Gesture Controlled Robocar.

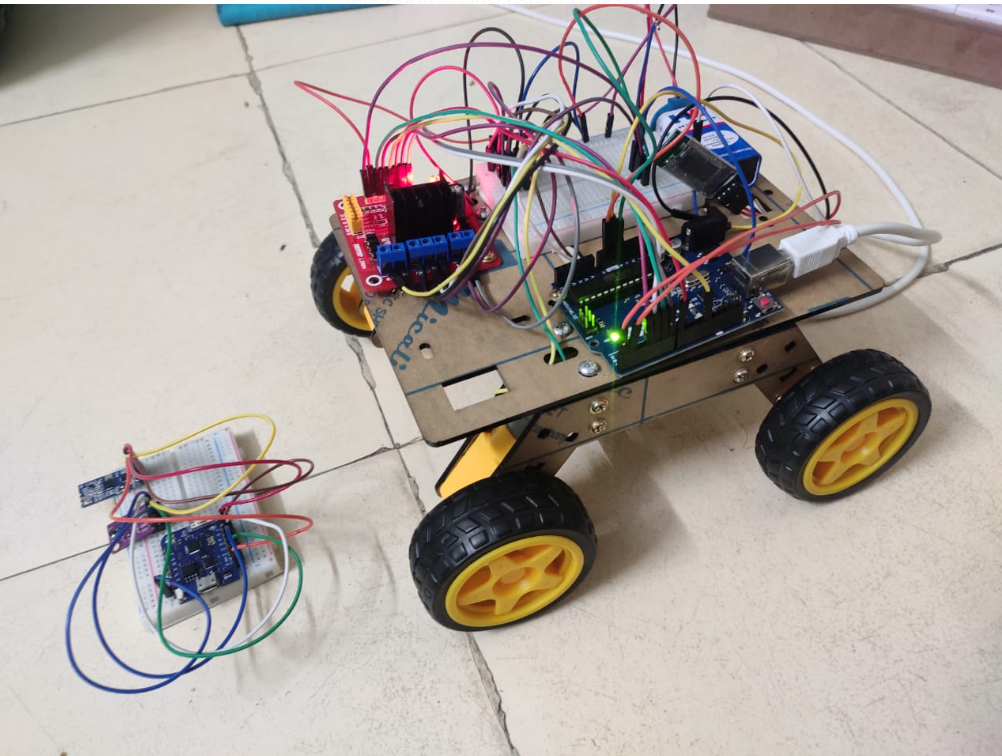
We will do coding on Arduino IDE in the C language.

We will also attach some different color lights to the robocar car to give it a cool look.

**3.3 Images of final Projects**





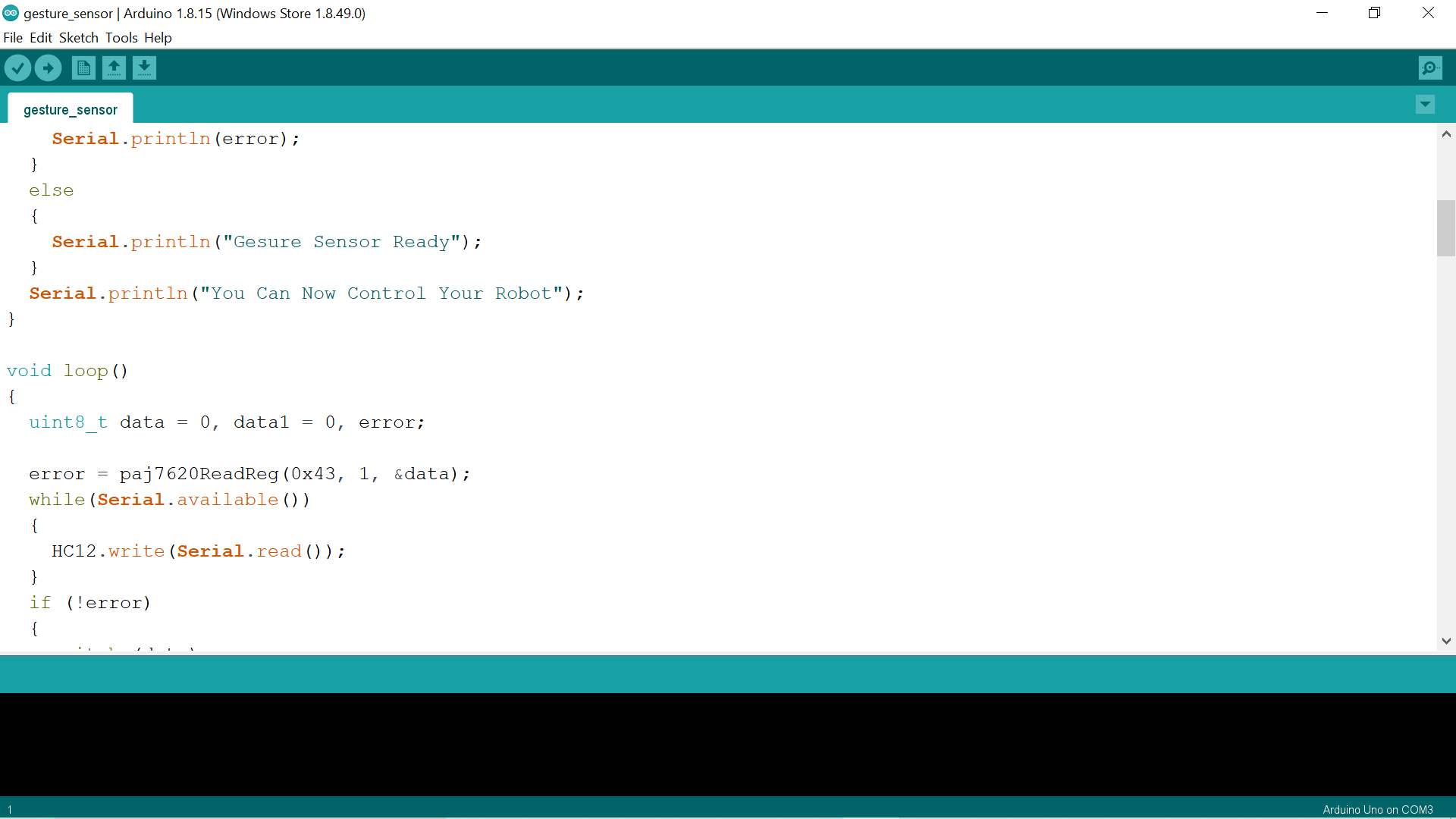
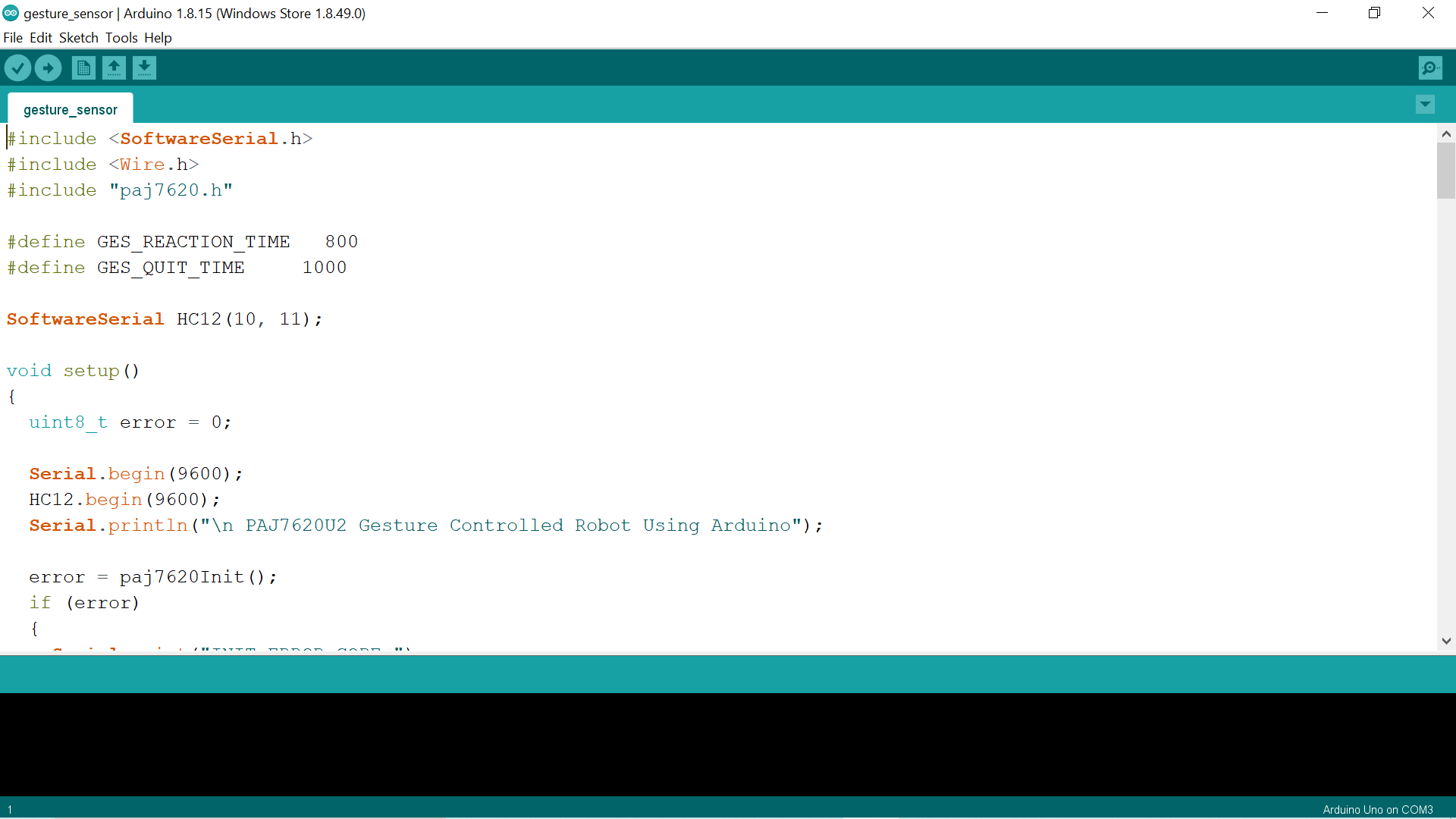


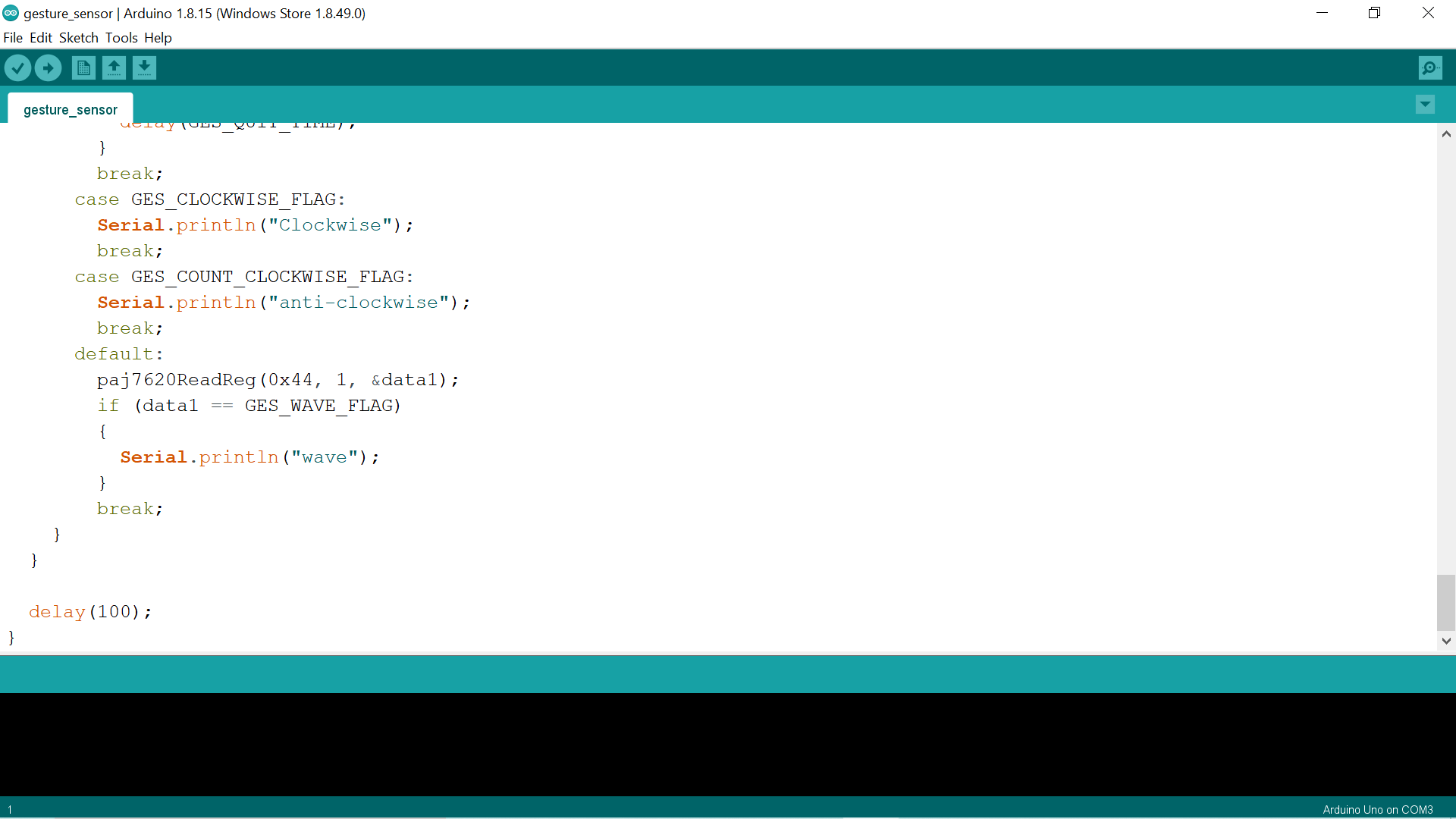
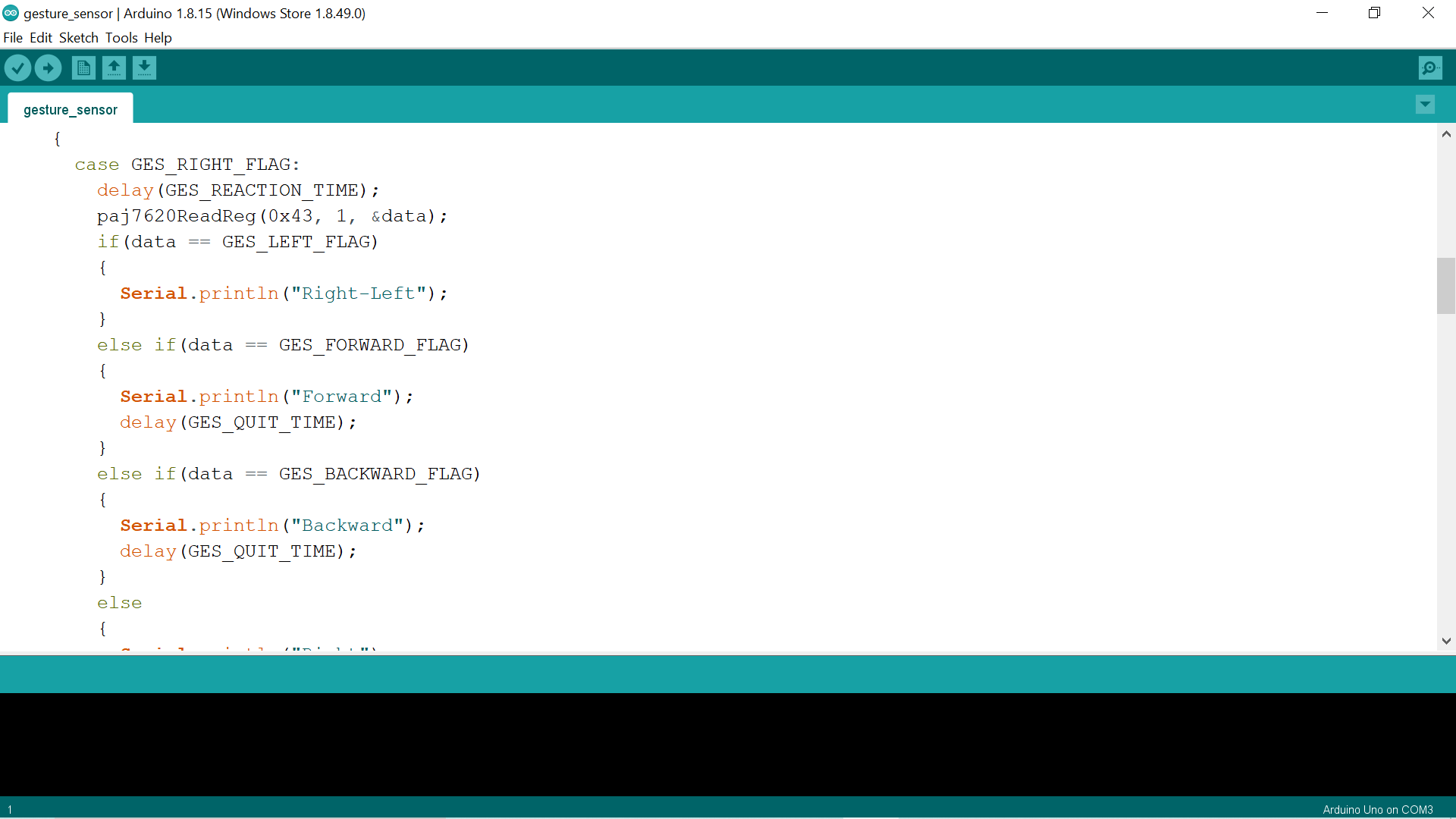
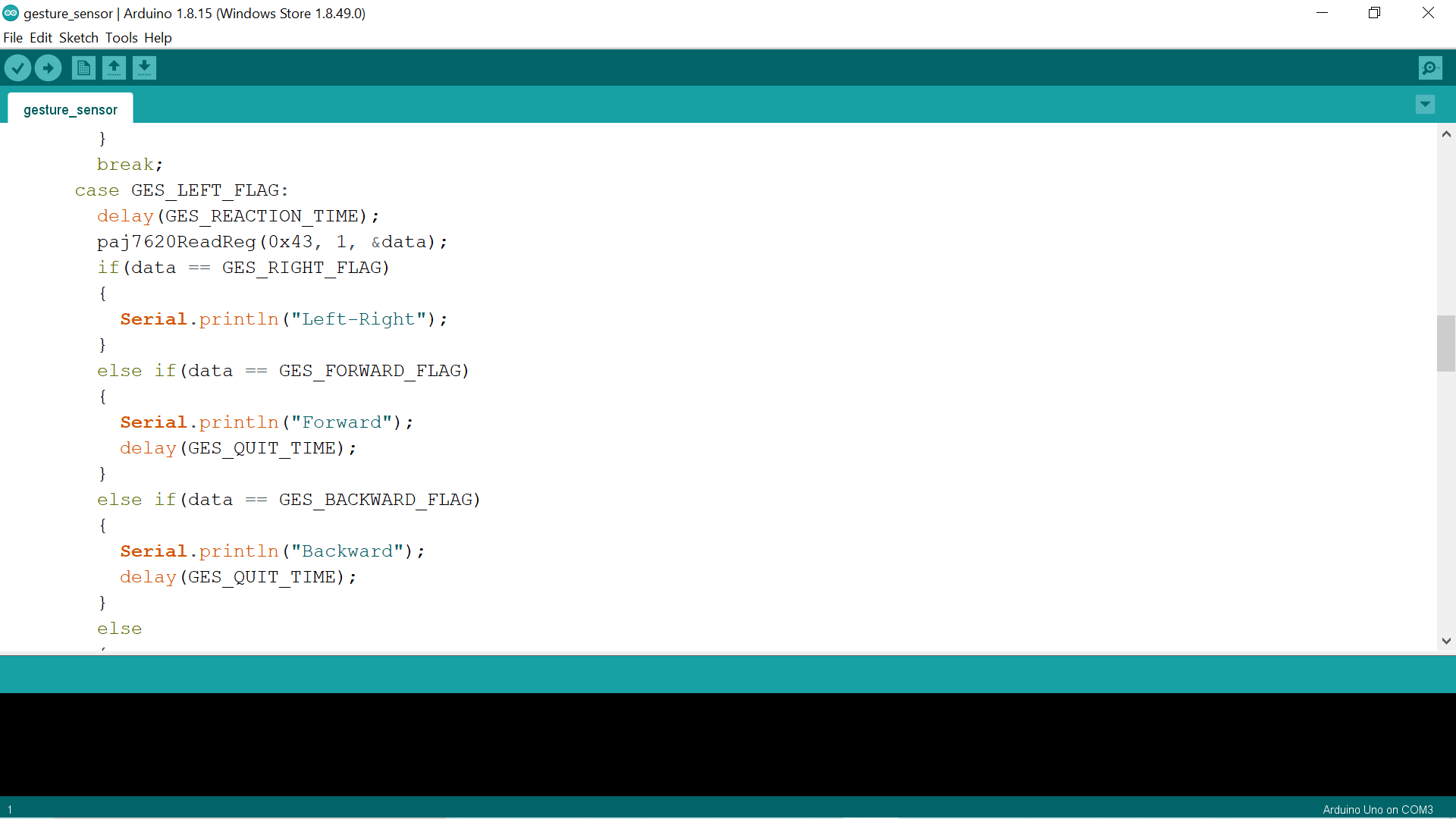
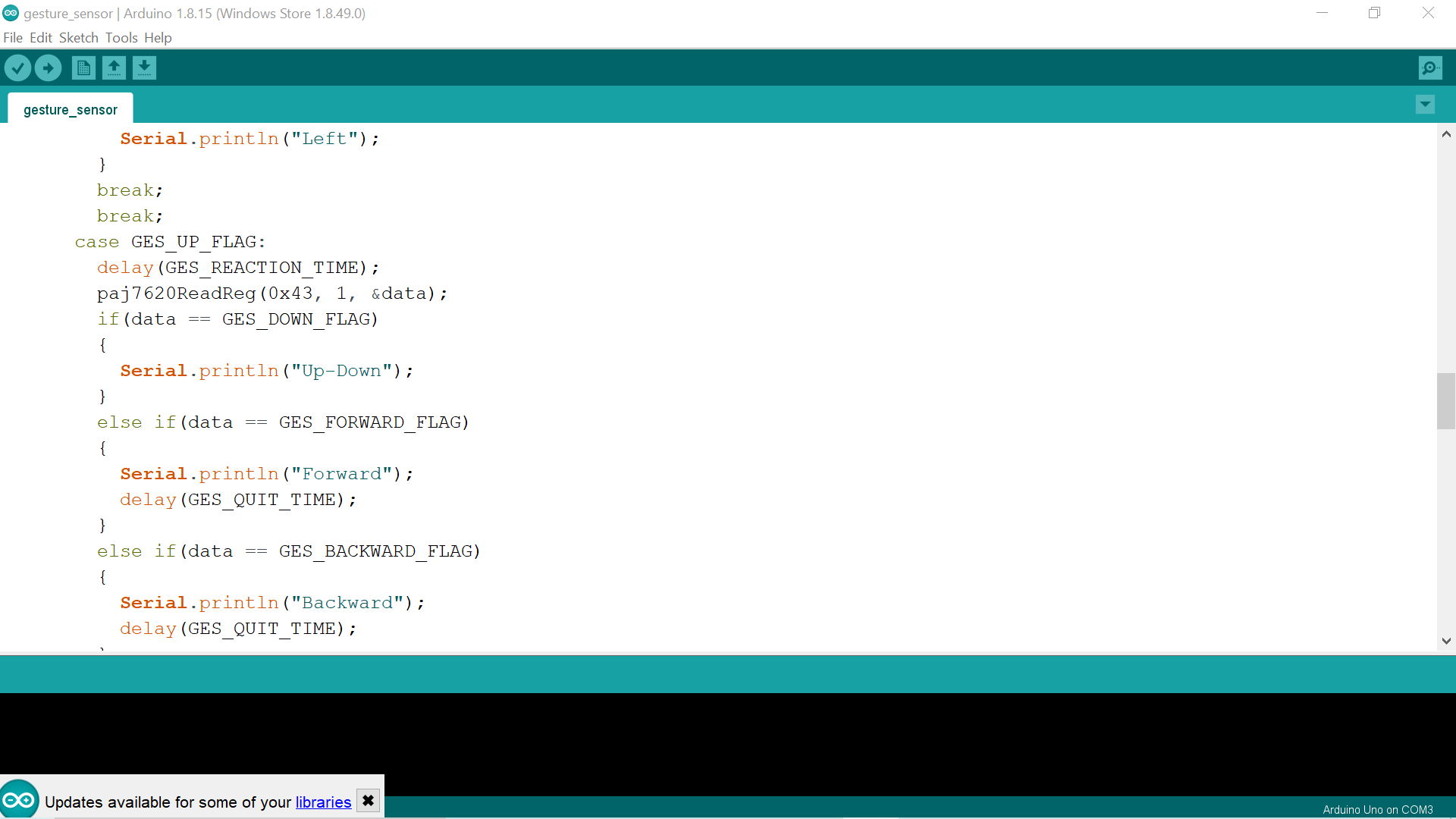
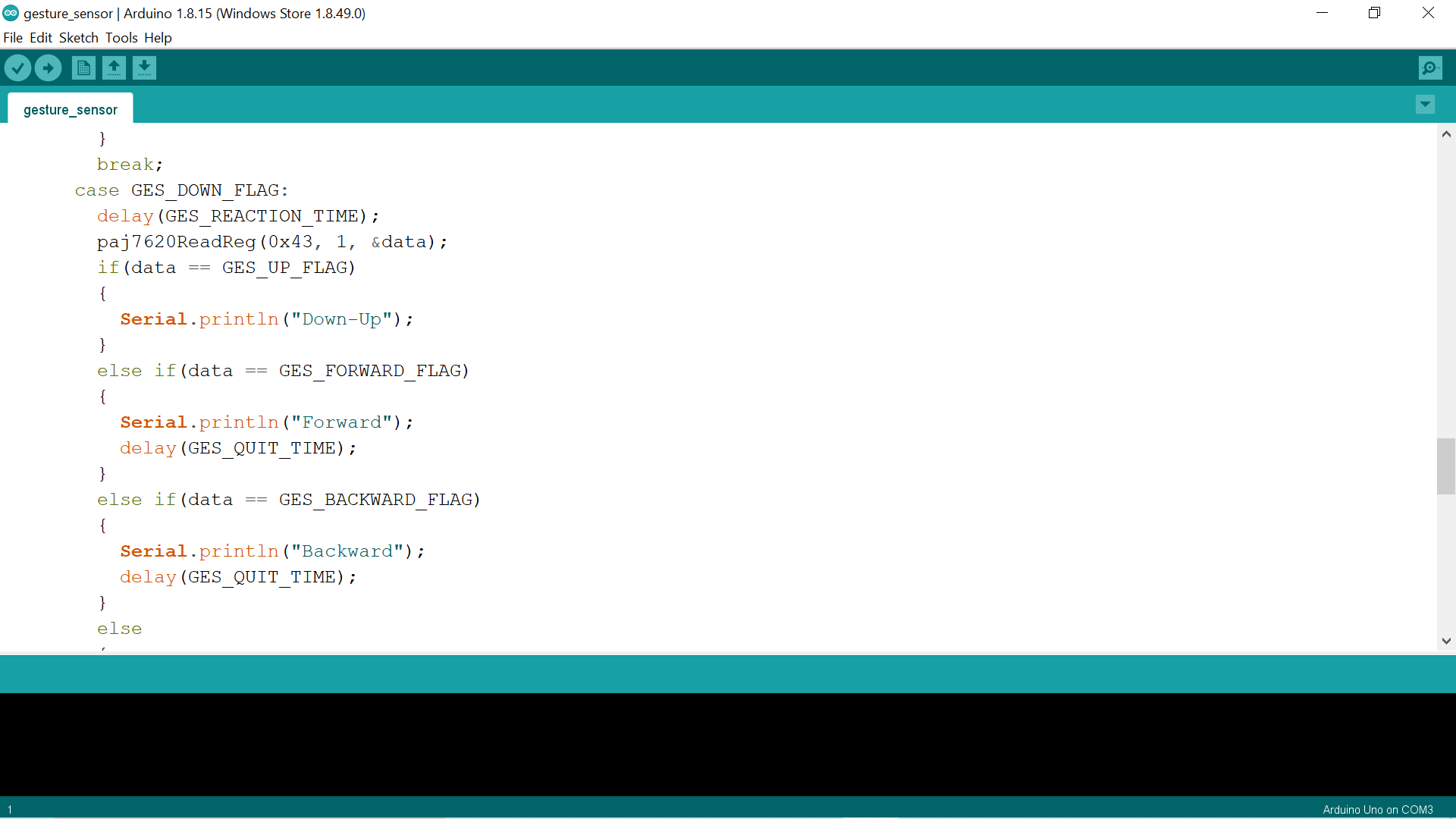
**CHAPTER 4**

**Software Development**

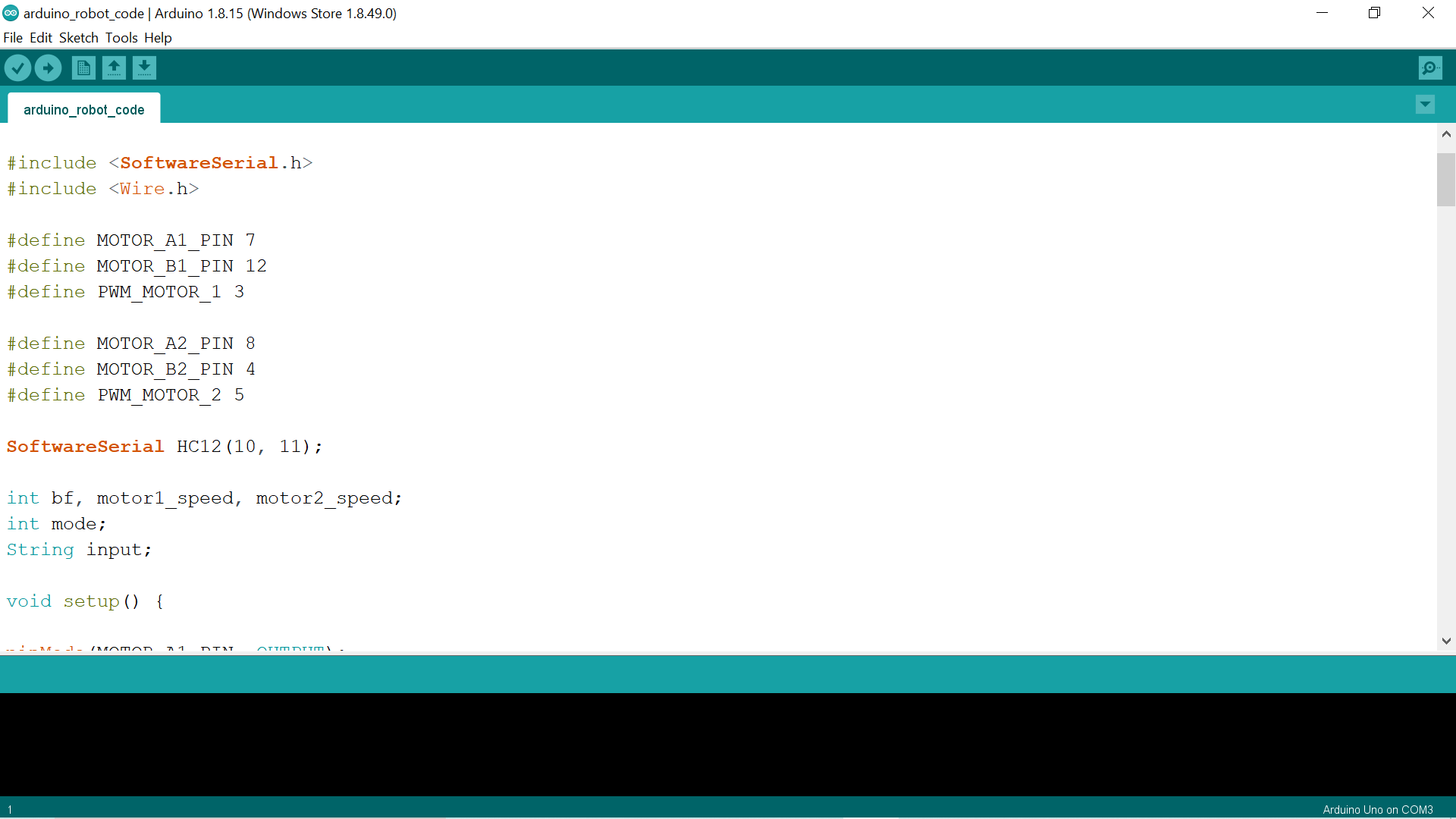
**Code:**

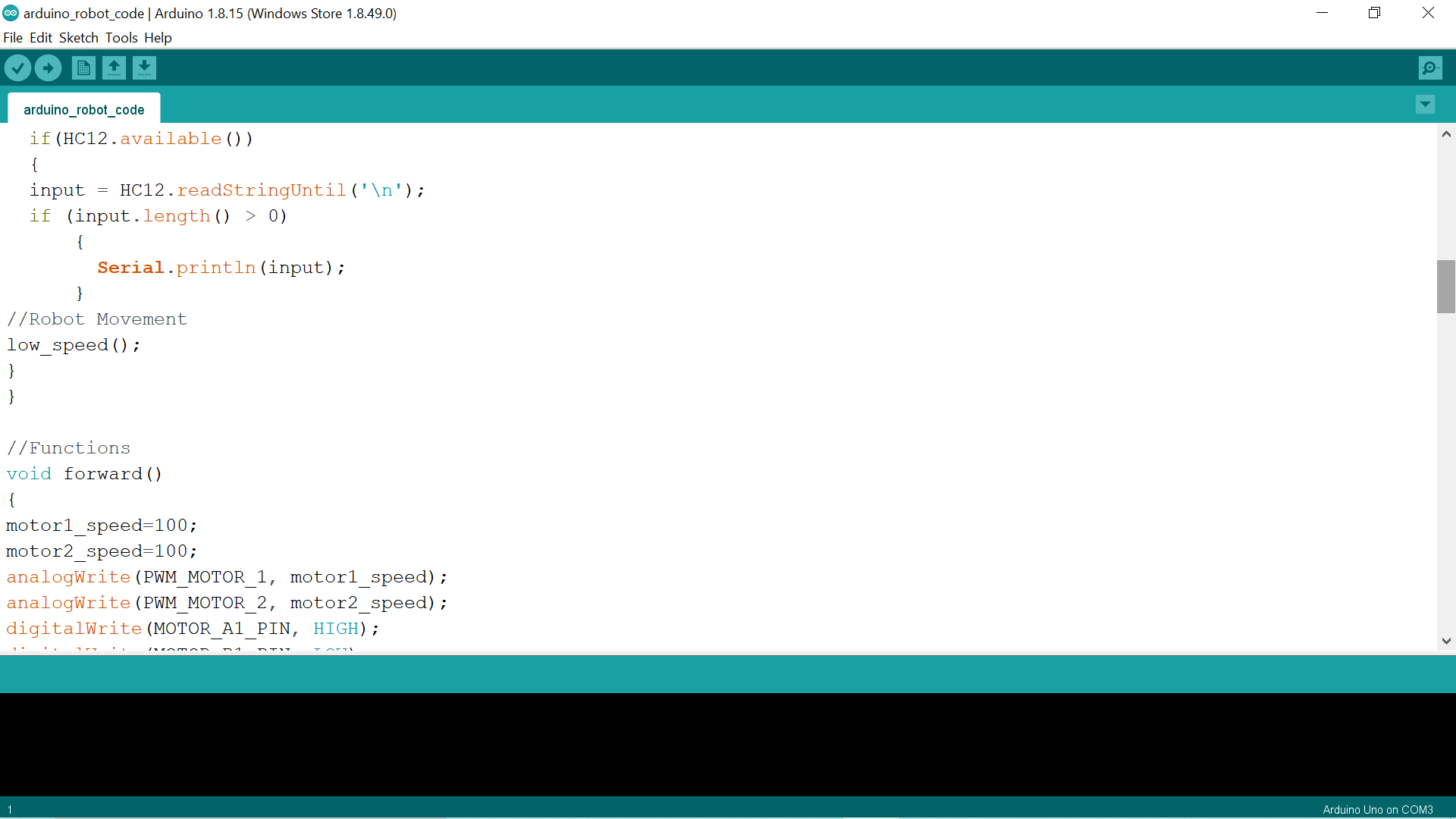
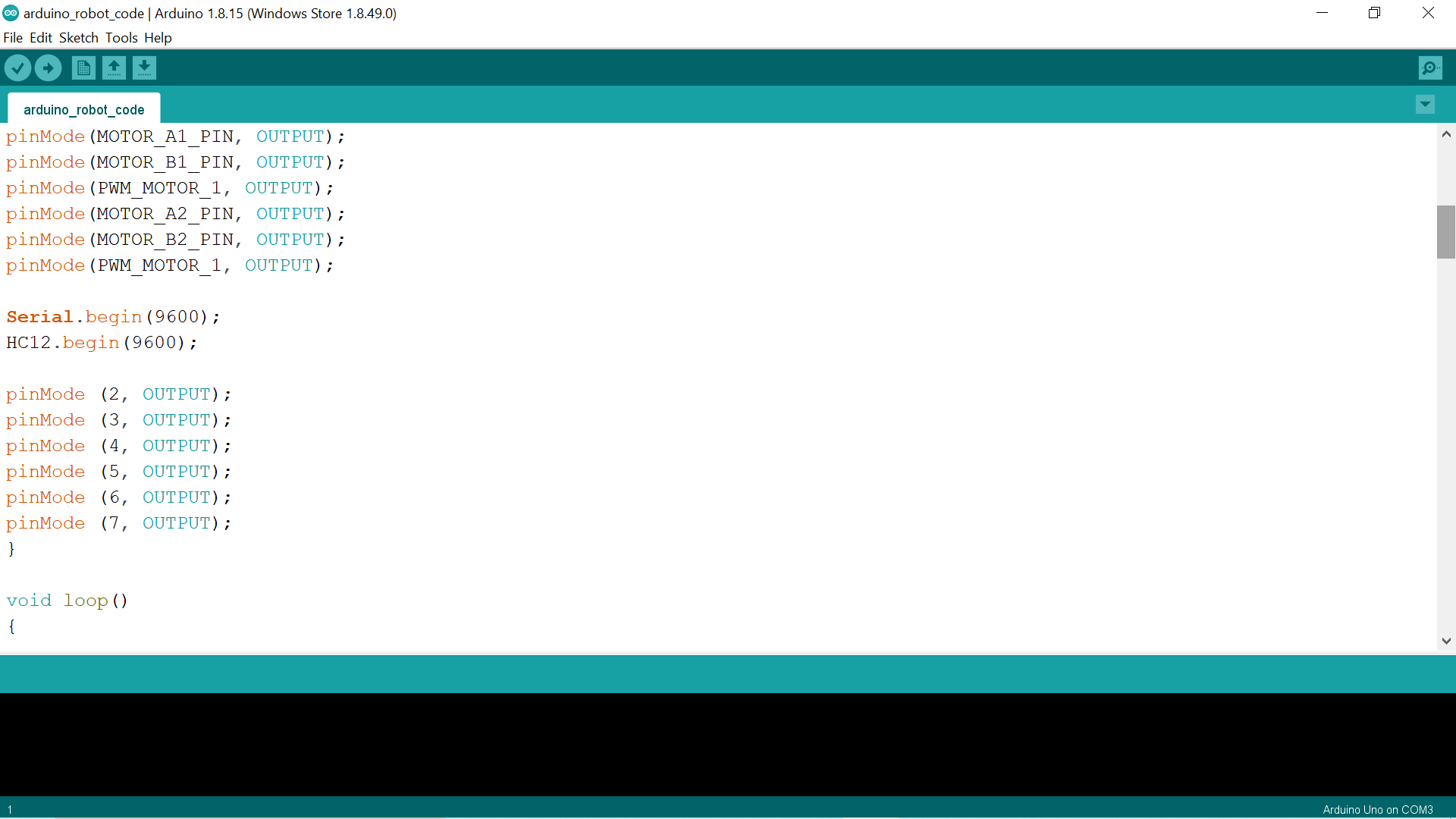
**Code for transmitting part:**

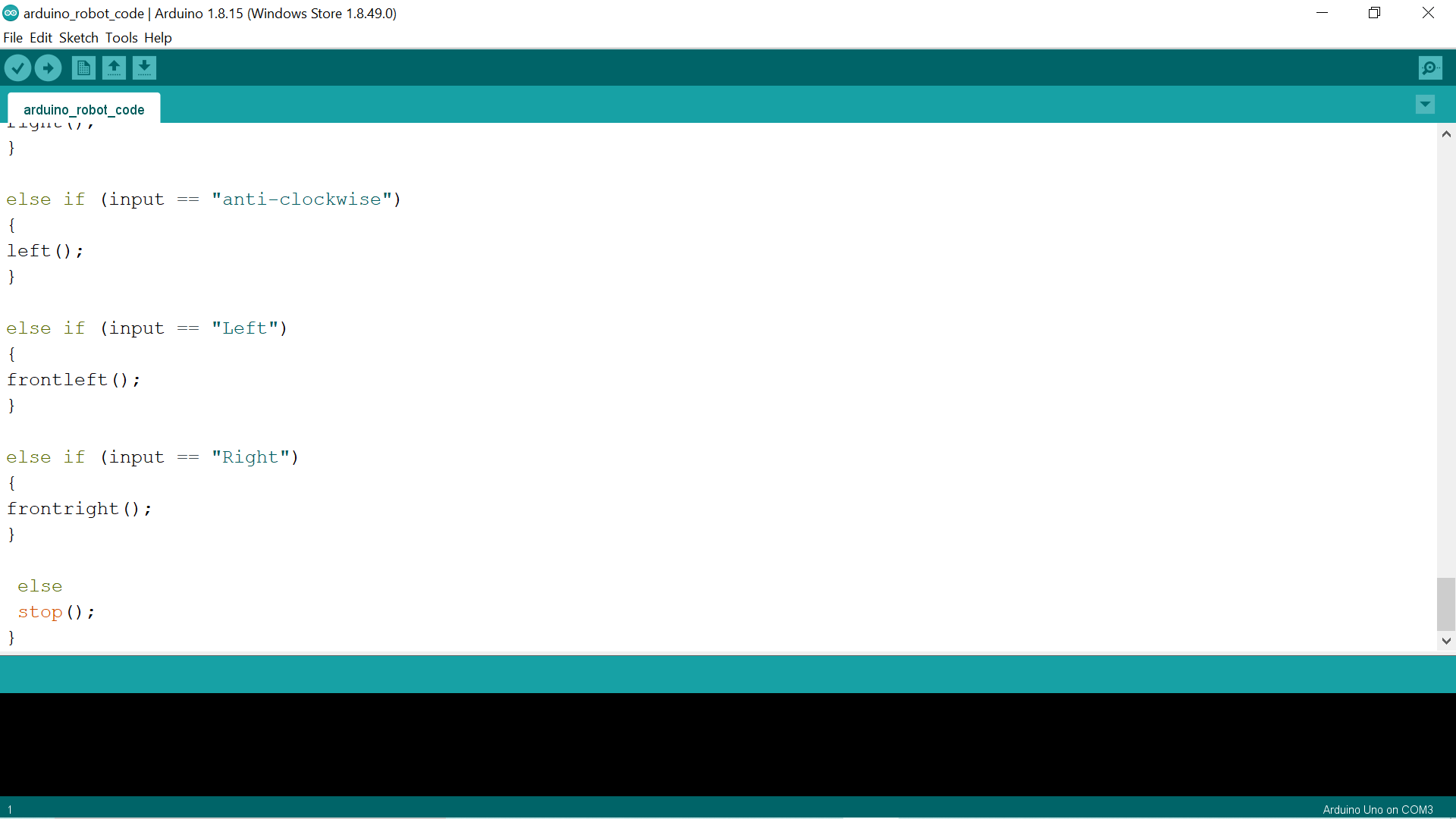
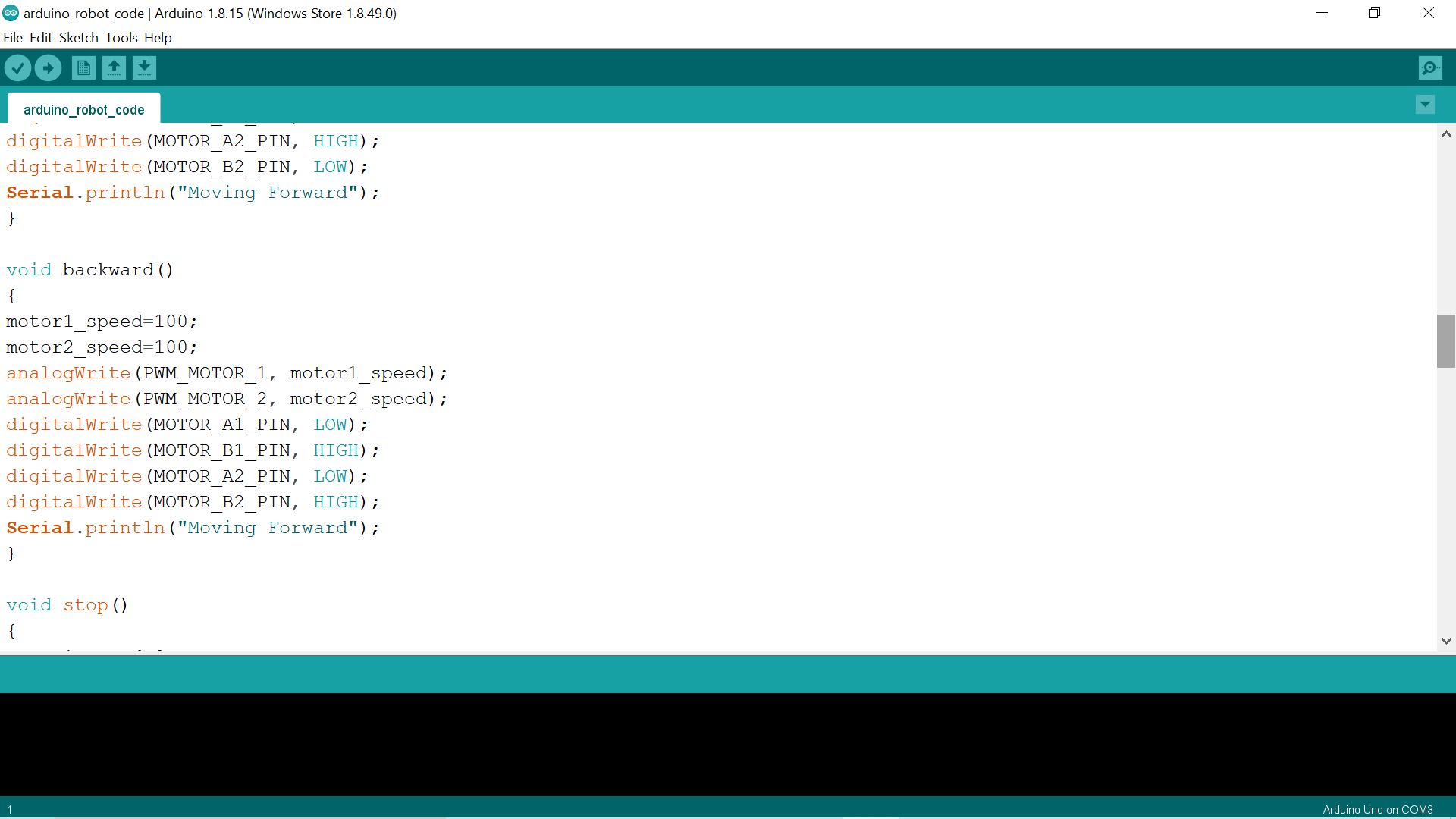
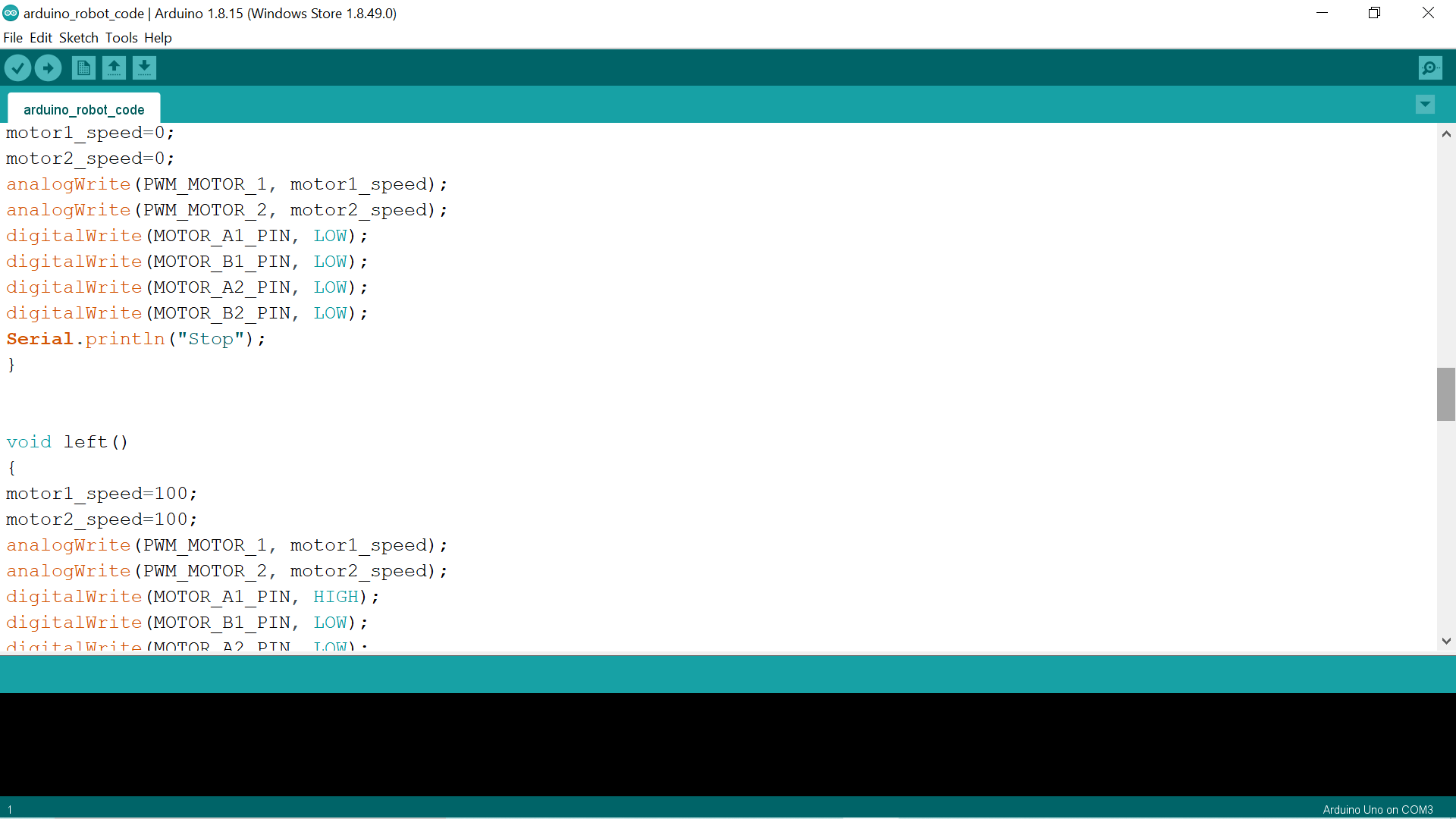
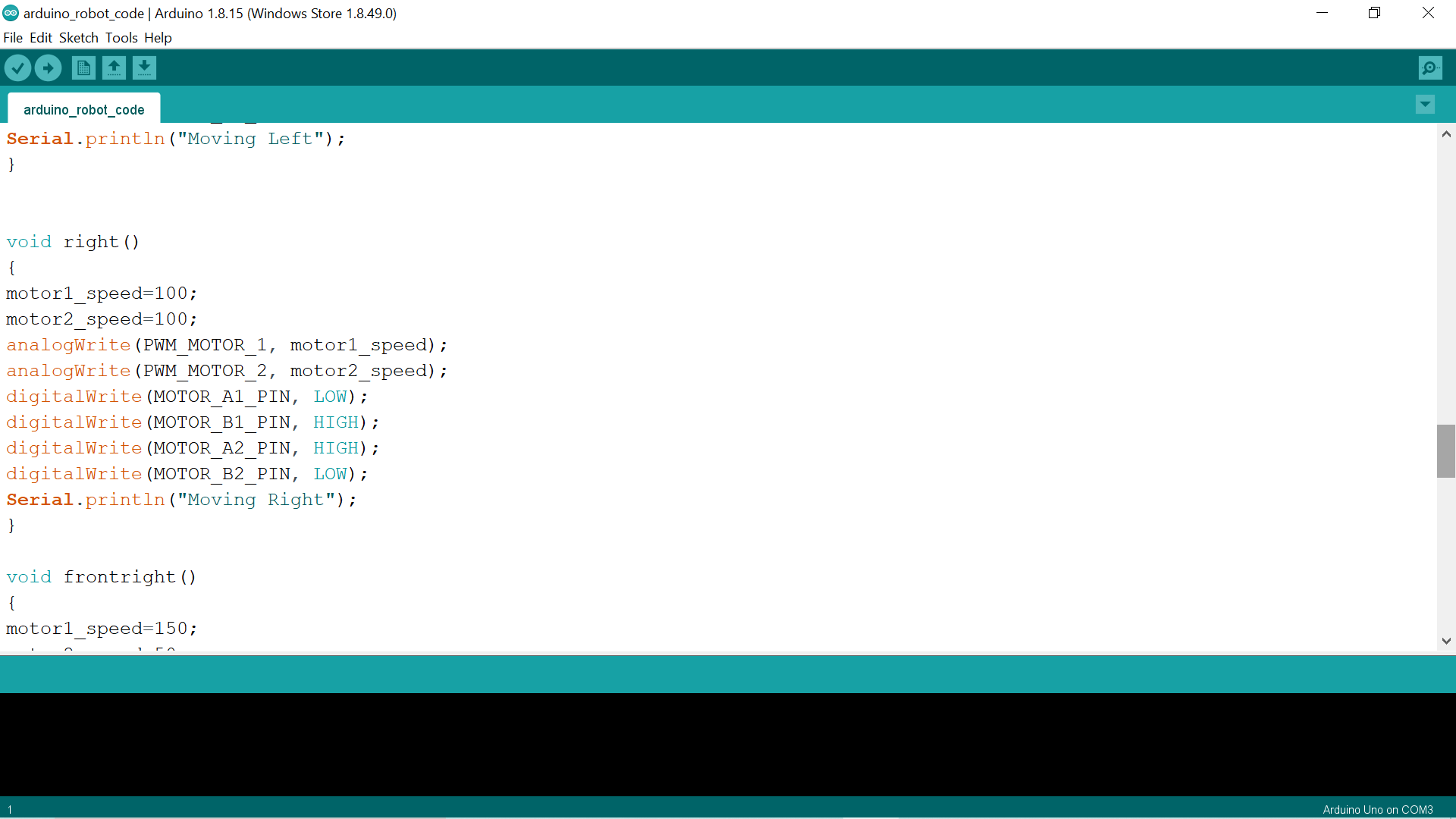
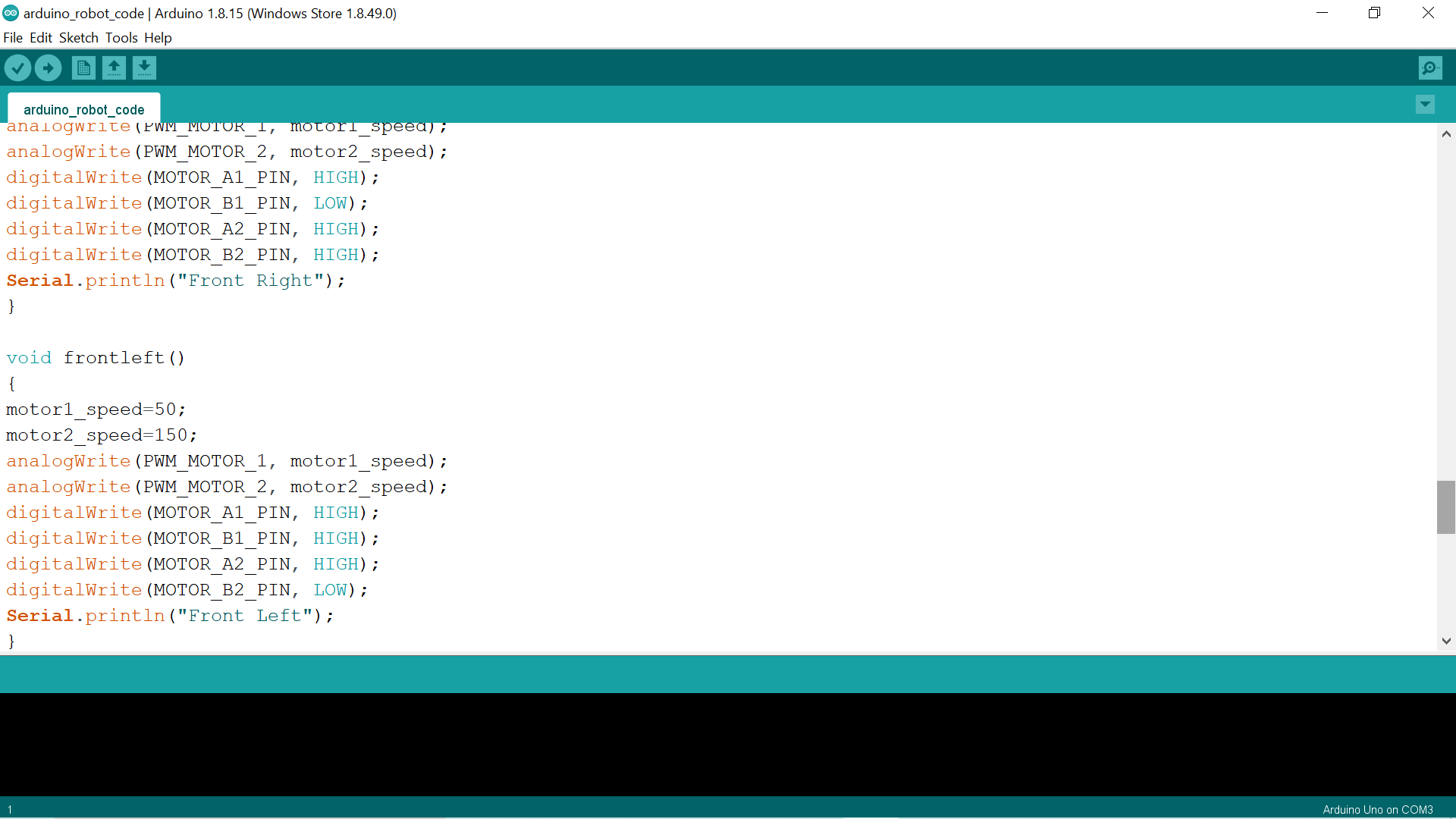
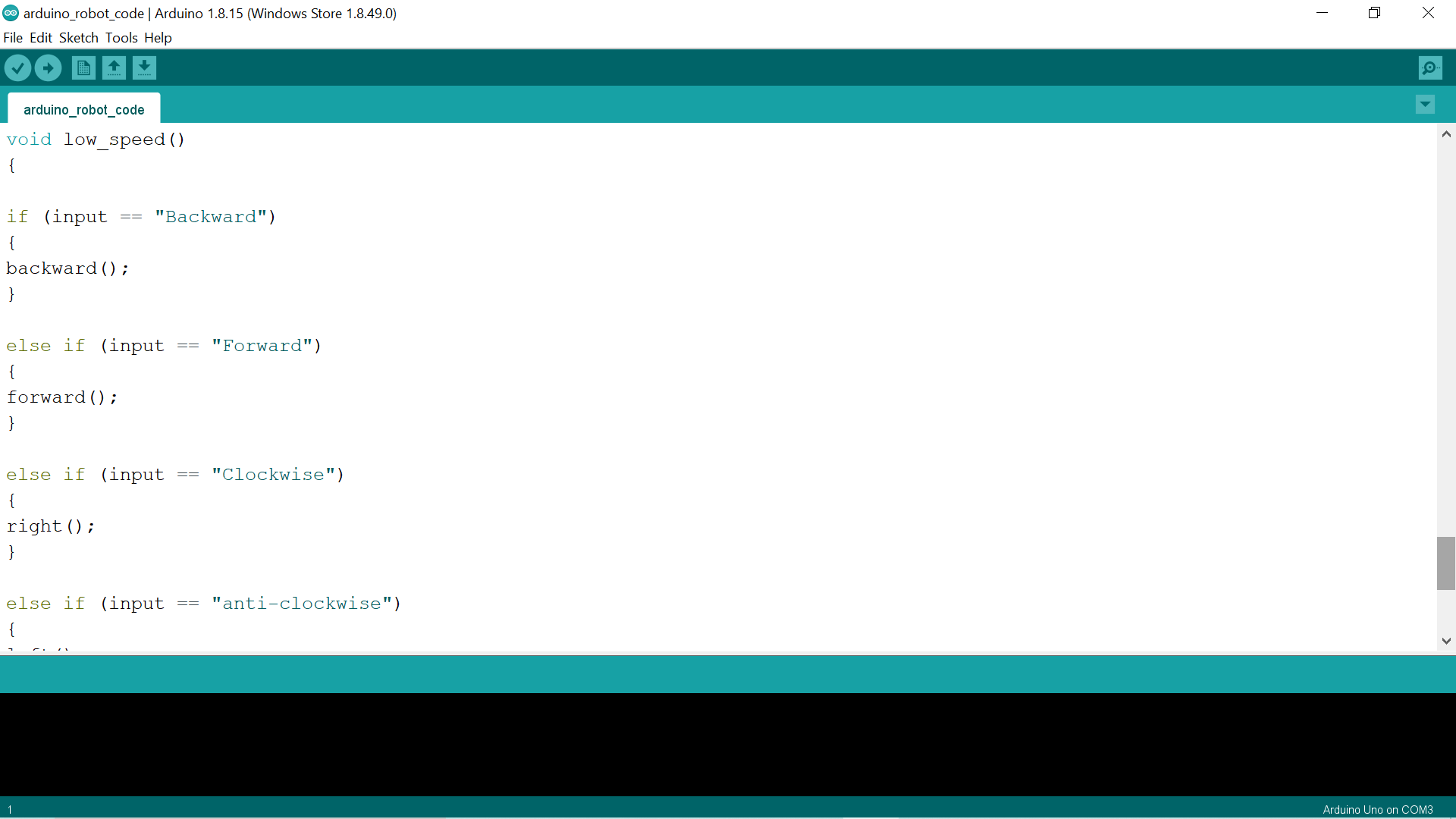




**Code for receiving side:**







**CHAPTER 5**

**Result and Conclusion**

**5.1 Result**

The robocar is moving according to the gestures we are giving to the sensor. Our gesture sensor can recognize 9 different gestures and we can move our robocar according to it. The processing of the data on transmitting side is done by Arduino UNO and on receiving side also it is done by Arduino Uno.

**5.2 Conclusion**

First, we had given the gesture to the gesture sensor. The gesture sensor then transferred the data to the Arduino UNO for the processing. With the help of HC-12 Wireless module we transferred this data to the HC-12 Wireless module on the receiving side which will receive this data and give this data to Arduino UNO for processing. Then Arduino uno will give signal to L298 bridge motor driver and it will move the motor according to it.

**References**

1. [V6I6-0180.pdf](file:///D:\IP%20project\robo%20papers%20by%20ayush\V6I6-0180.pdf)

2. [15-alonso-hri-icra.pdf](file:///D:\IP%20project\robo%20papers%20by%20amit%20sir\15-alonso-hri-icra.pdf)

3. [19\_Accelerometer.pdf](file:///D:\IP%20project\robo%20papers%20by%20ayush\19_Accelerometer.pdf)

4. [cvbmICRA2017\_final\_submission.pdf](file:///D:\IP%20project\robo%20papers%20by%20amit%20sir\cvbmICRA2017_final_submission.pdf)