# Gesture controlled RC car

Cristescu Vlad-Andrei

ETTI, group 2031

## Table of Contents

## Introduction

In this project I will attempt to modify a store bought remote controlled car to be able to control it by tilting my hand. The goal is to make the car go in the direction in which I tilt my palm. My approach to this project is to use as much of the already existing circuitry on the car, so I will simply connect to the transmitter and use the signals sent from a sensor instead of pressing the buttons on the remote. For this I will need the following:

# Hardware Components

**1. A toy RC car and remote**

This is a cheap, generic toy car.



**2. Arduino Uno**
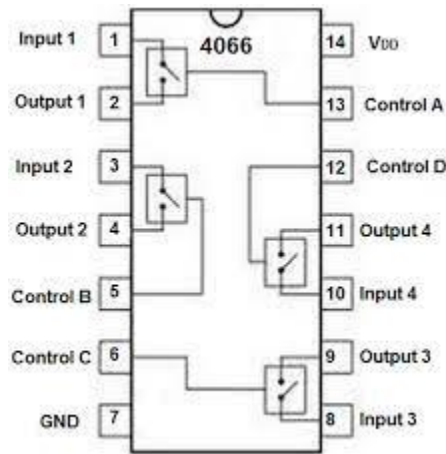This board contains a microcontroller which is going to be the "brains" of the circuit.



**3. MPU-6050**
This module contains both an accelerometer and a gyroscope.

## 4. CD4066

This integrated circuit allows the Arduino to simulate pressing the control buttons on the remote. It has 4 channels controlled by 4 switches:



# Theoretical Background

We are looking for a device that can sense the inclination of the palm and turn it in a signal which we can measure, process and send to the car. The MPU-6050 is a MEMS (micro-electro-mechanical sensor) which can turn a mechanical stimulus in an electrical signal.
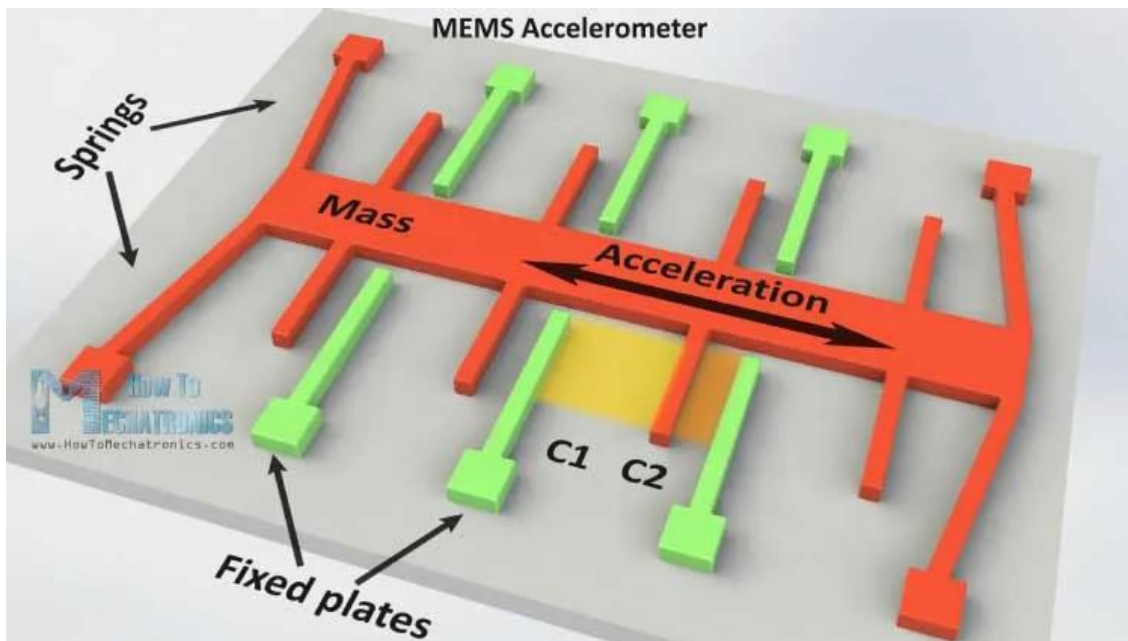


Changing the distance between the plates by accelerating a mass creates a change in the capacitance. This can be measured in order to determine the orientation and the force of the acceleration. This signal will then be processed by the Arduino using software to control the digital outputs which are connected to the control pins of the CD4066 . In this manner we manage to "press the buttons" on the remote according to the readings of the gyroscope.

## Circuit Connections

The gyroscope is connected to the Arduino in order to collect data. This then sends appropiate instructions to the 4066, commanding which paths to close on the PCB of the remote.
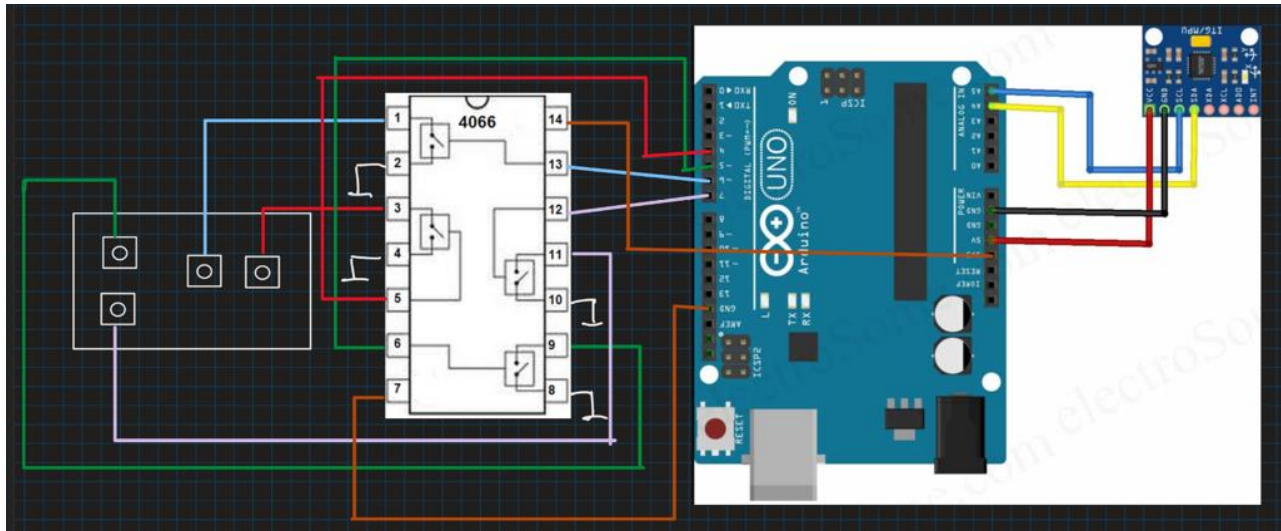


## Code Comments

The purpose of the software is to process the data from the gyroscope and to control the switches. First we have to define the register addresses and some variables for the data. In the setup section we have to wake up and put the sensor in normal mode and also select the sensitivity of the sensor.

In the loop section we will read the data for the X, Y and Z axis. Then the raw data has to be converted into angle values. From the datasheet of the sensor we can see that for the 2000dps sensitivity mode corresponds a 70 mdps/digit unit. This means that we have to multiply the raw output data by 0.07 in order to get the angular rate in degrees per second. Then if multiply the angular rate by time it will give us the angle value. So we need to calculate the time interval of each loop section and we can do that by using the millis() function at the top and the bottom of the loop section, and we will store its value into this "dt" variable. So for each executed loop we will calculate the angle and add it to the final angle value.

We will do the same for the two other axis and at the end we will print the results in the serial monitor. For this specific application we only need 2 axis, X and Y. If the angle of the X-axis is positive, the car must go forward, if it's negative it must go backwards. Similarly the angle of the Y-axis determines the left/right movement.

## Conclusions

This is a fun mini-project with lots of space for improvement. There are a lot of functionalities one could add but the goal of this project was merely to modify a RC car in order to control it by tilting my hand, which was achieved.

## Bibliography

https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/

https://www.brainy-bits.com/post/using-the-4066-ic-bilateral-switch-to-simulate-pushing-buttons

https://mschoeffler.com/2017/10/05/tutorial-how-to-use-the-gy-521-module-mpu-6050-breakout-board-with-the-arduino-uno/

https://www.alldatasheet.com/view_datasheet.jsp?Searchword=MPU-6050

https://www.alldatasheet.com/view_datasheet.jsp?Searchword=CD4066