Checkpoint 2　Report

[ICN5406] Mobile Robot 2018

Student ID: 0745011　**Name: Petr Gondek**　**Date: 25/10/2018**

1. Purpose:

Task for the second checkpoint was make the robot able to move. We needed to install two wheels with two motors, one support wheel, H-Bridge board, connect components to Arduino and code program which controls the movement of the robot.

2. Description of Design:

We installed components as you can see on figure 1. Two wheels are attached to two DC motors and motors are connected to H-Bridge (red/black wires). We connected a power supply (black/red/white wires).

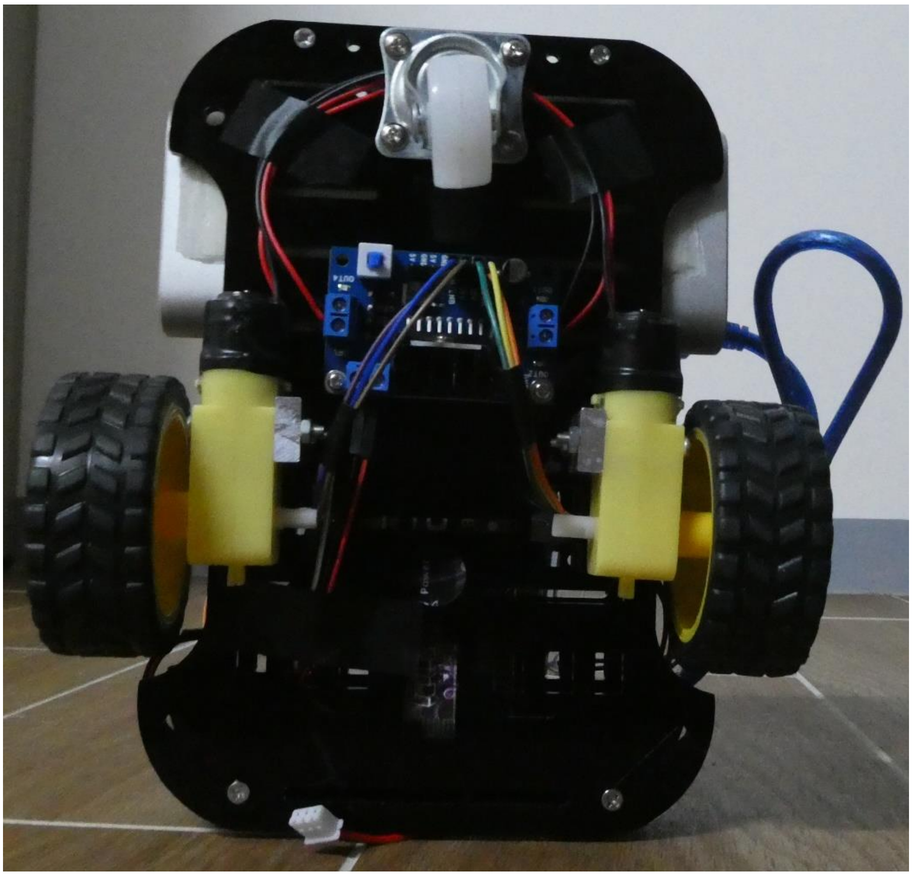


Figure 1 – Setup

As next step we connected Arduino an H-Bridge via colored wires. Orange, yellow and green control left wheel and brown, blue, and purple control right wheel. The orange and the purple wires are used for PWD signal. They control angular velocity. On figure 2 and figure 3 you can see how they are connected.

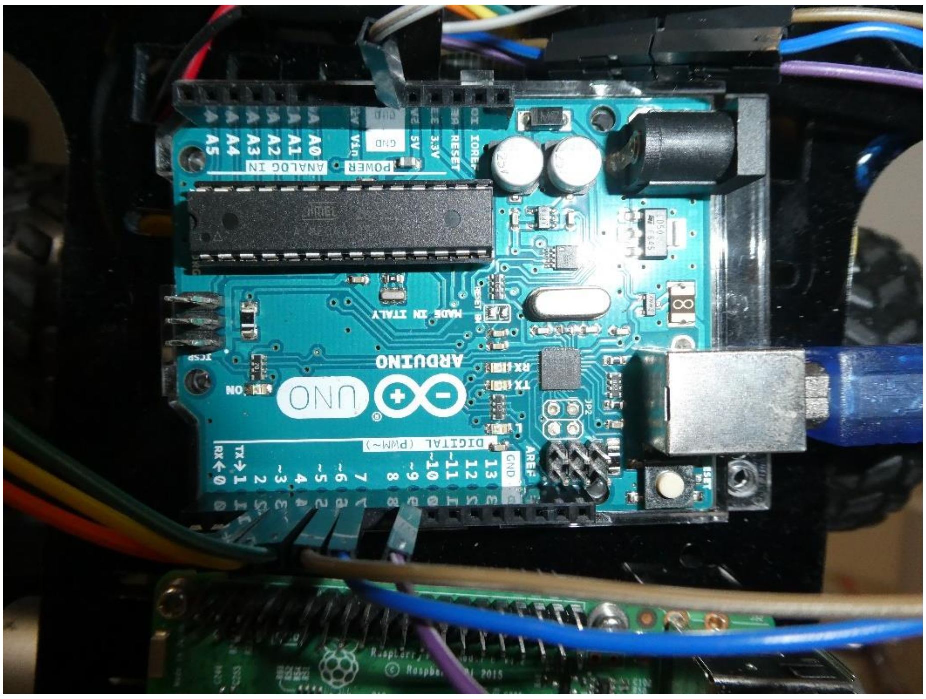


Figure 2 - Arduino

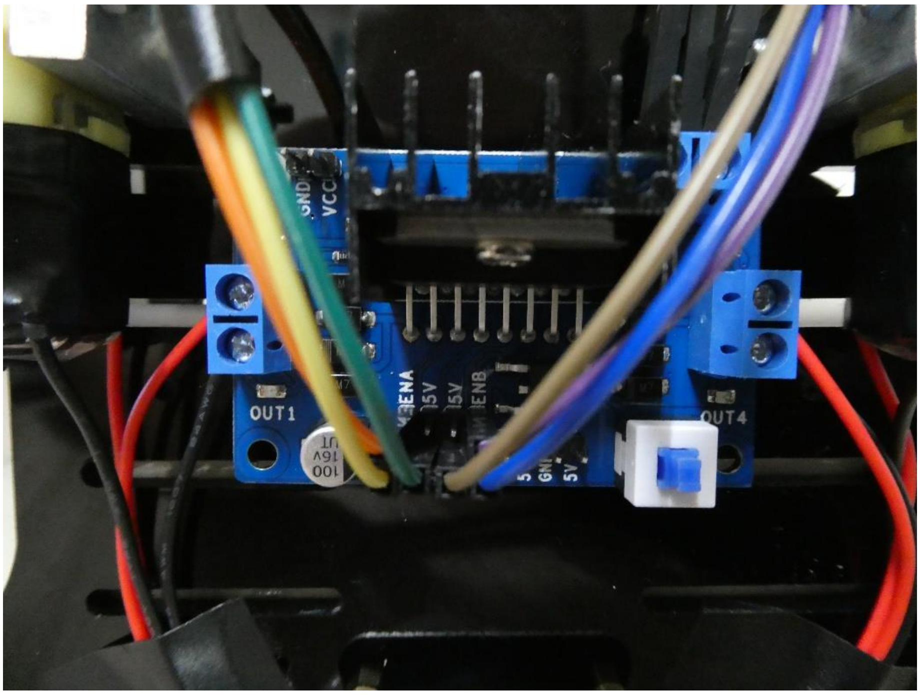
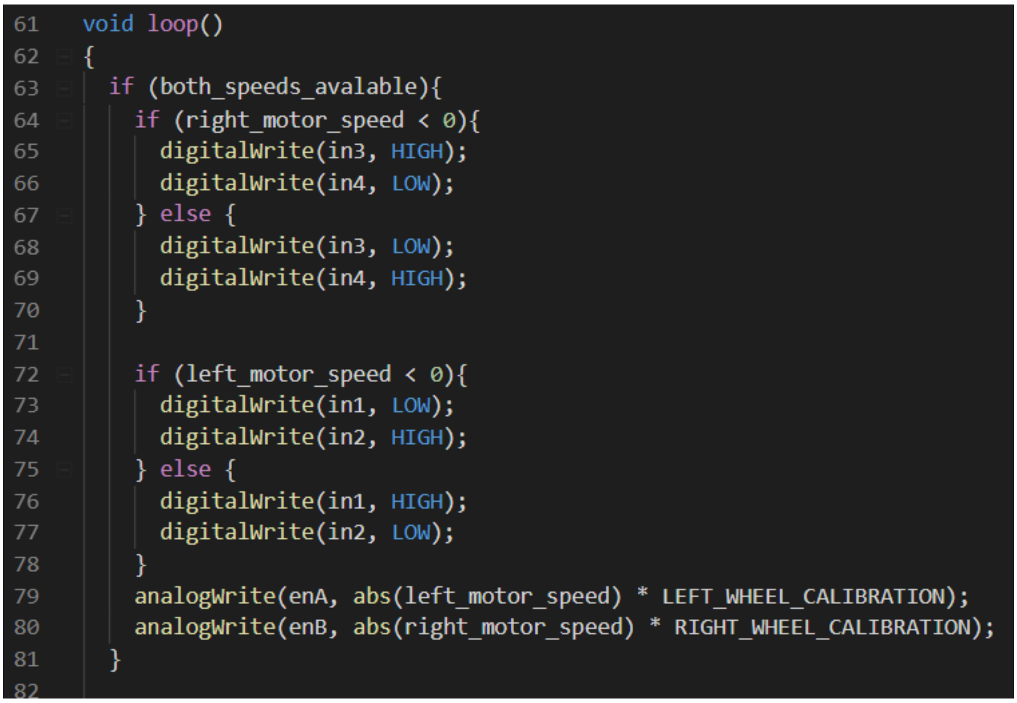


Figure 3 - H-Bridge

We coded simple publisher for Raspberry PI. Publisher sends value [-255; 255] to Arduino. When Arduino receives messages for both wheels:

- Arduino detects direction and sends to corresponding pins HIGH or LOW signal.
- Arduino sends PWM speed signal to corresponding pins multiplied by calibrations constants.



```
void loop()
{
  if (both_speeds_avalable){
    if (right_motor_speed < 0){
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
    } else {
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
    }

    if (left_motor_speed < 0){
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
    } else {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
    }
    analogWrite(enA, abs(left_motor_speed) * LEFT_WHEEL_CALIBRATION);
    analogWrite(enB, abs(right_motor_speed) * RIGHT_WHEEL_CALIBRATION);
  }
```

Figure 4 - Arduino code

For moving forward, we need to send on `in3` and `in2` `HIGH` value and `in4` and `in1` `LOW` value. If we switch values the robot will move backward.

Calibrations constants was pick based on test drive. We chose initial speed. Then we found two speeds (by increasing speed for one wheel and decreasing speed for the other) when robots move directly. After that we calculated percentual change from initial speed.

3. Result

On final contest our robot didn't moved straight forward. Before final drive we had problem with our wheels. During one test drive we had 1.2:0.8 calibration's constants and during second drive we had opposite values – 0.8:1.2. We were unable to stabilize the robot. Every test drive the constants were totally different. We don't know why this happened.

4. Discussion

It is almost impossible to make robot move straight forward with cheap motors and without sensors. There are so many factors which can change direction of the robot. For example, the quality of motors, power supply and position of back wheel. Because of this, the part that the robot can’t move straight forward should be less rated.