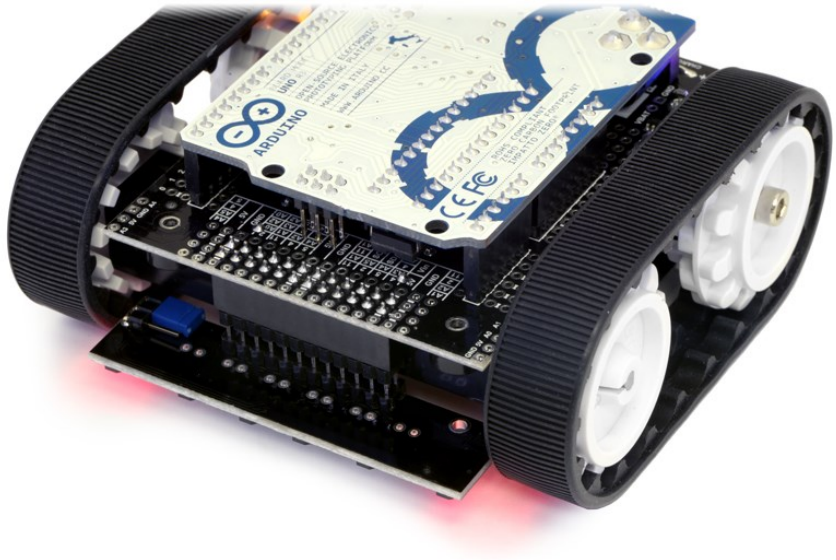
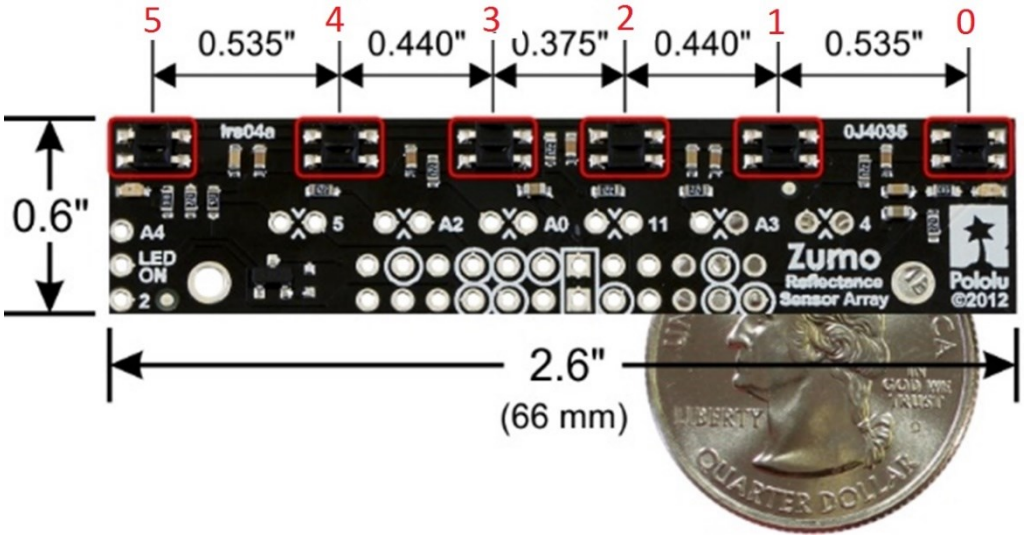


Двигатели

Сигналите за управление на двата двигателя се извеждат от 4 цифрови пина на Arduino, които вече са свързани с разширителната платка ZumoShield:

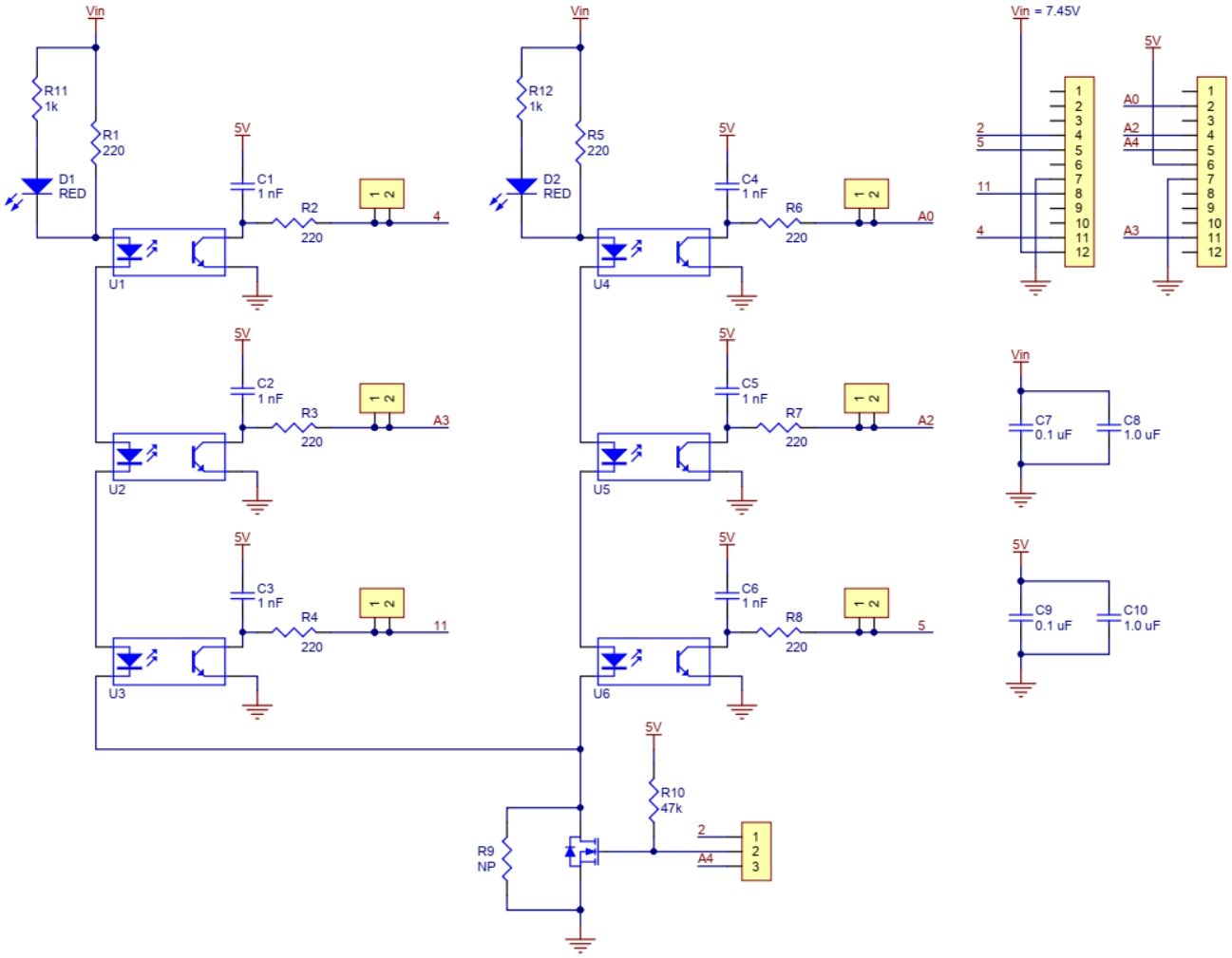
- **пин 7** задава посоката на десния двигател (LOW напред, HIGH назад – ако са правилно запоени);
- **пин 8** задава посоката на левия двигател;
- **пин 9** задава скоростта на десния двигател (0 – 255);
- **пин 10** задава скоростта на левия двигател (0 – 255).



Pololu
Robotics & Electronics

Schematic Diagram for the Zumo Reflectance Sensor Array

www.pololu.com



Функции

ZumoReflectanceSensorArray

void readCalibrated(unsigned int *sensorValues, unsigned char readMode = QTR_EMITTERS_ON)

Returns sensor readings calibrated to a value between 0 and 1000, where 0 corresponds to a reading that is less than or equal to the minimum value read by calibrate() and 1000 corresponds to a reading that is greater than or equal to the maximum value. Calibration values are stored separately for each sensor, so that differences in the sensors are accounted for automatically.

---

unsigned int readLine(

unsigned int* sensorValues, unsigned char readMode = QTR_EMITTERS_ON, unsigned char whiteLine = 0

)

Връща 0, ако линията е под сензор 0; 1000, ако е под сензор 1 и т.н. Ако линията е едновременно под сензори 2 и 3 (т.е. в центъра), readLine() връща 2500, защото (2000 + 3000) / 2 = 2500. Това е вашата желана стойност, така че трябва да компенсирате отклонението от нея – чрез двигателите.

Operates the same as read calibrated, but with a feature designed for line following: this function returns an estimated position of the line. The estimate is made using a weighted average of the sensor indices multiplied by 1000, so that a return value of 0 indicates that the line is directly below sensor 0 (or was last seen by sensor 0 before being lost), a return value of 1000 indicates that the line is directly below sensor 1, 2000 indicates that it's below sensor 2, etc. Intermediate values indicate that the line is between two sensors. The formula is:

$$\frac{0*value0 + 1000*value1 + 2000*value2 + ...}{value0 + value1 + value2 + ...}$$

As long as your sensors aren't spaced too far apart relative to the line, this returned value is designed to be monotonic, which makes it great for use in closed-loop PID control. Additionally, this method *remembers where it last saw the line*, so if you ever lose the line to the left or the right, its line position will continue to indicate the direction you need to go to reacquire the line. For example, if sensor 4 is your rightmost sensor and you end up completely off the line to the left, this function will continue to return 4000.

By default, this function assumes a dark line (high values) surrounded by white (low values). If your line is light on black, set the optional second argument whiteLine to true. In this case, each sensor value will be replaced by the maximum possible value minus its actual value before the averaging.

ZumoMotors

void setSpeeds(int leftSpeed, int rightSpeed) – задава скоростта на левия и десния двигател

Pushbutton

waitForButton() – програмата спира и изчаква потребителят да натисне хардуерния бутон в разширителната платка Zumo

Задачи

1. Изпробвайте движения (вкл. назад) с различни скорости на колелата. Открийте максималната стойност на параметрите за скорост на левия и десния двигател.

2. Проверете какви стойности връща подредбата от ИЧ сензори при различни положения на робота върху линията и извън нея. За извеждане на стойностите използвайте функцията Serial.print() и инструмента Serial monitor на Arduino IDE.

3. Съставете и реализирайте алгоритъм за следене на линия. Роботът трябва да се движи непрекъснато, като поне един от двигателите му се върти с максимална скорост. Отклонението от линията се компенсира със съответно намаляване на скоростта на един от двата двигателя (другият запазва максималната си скорост).

Начален код:

в прилежащия файл zumo.ino