

Motor Programming

Smetankin Daniel

Intro

In this report the programming of the motor of the car will be divided into different sections and explained.

Discussion & Conclusion

The circuit is based on ESP8266, and the software is written in the Arduino IDE. We didn't open the car yet so we don't know what kind of motor powers the car. But we will assume it is a standard servo motor attached to a PWM to regulate the vehicle's speed. A value of 0 to 1023 can be send to the PWM witch will control the cars speed.[1]

1. Defining

Here are de most important variables defined.

```
#define PIN_MOTOR_A_IA D2
#define PIN_MOTOR_A_IB D3

#define PIN_MOTOR_B_IA D5
#define PIN_MOTOR_B_IB D6

int motor_speed;
bool motor_dir;
int force_turn;
```

2. MotorSpeed

Here servomotor A is configured which is responsible for moving the car back or forward. MotorSpeed will read an integer `m_speed` which can have a value between 0 and 1023. With 1023 a value that represents the full power of the servomotor. `Motor_dir` is a parameter which is used to check in what direction the car wants to move, straight ahead or reverse. The Boolean `dir == true`, when the car wants to drive straight ahead and `dir == false` when the car wants to reverse. The if statement checks if `dir == true`, if it is it will write an analogue value of 1023-PWM to input D2, and it will send a digital 1 or HIGH to input D3. If it is not true it will send the analogue value of the variable `pwm`, and a digital 0 or LOW.

```
void MotorSpeed(int m_speed){
    MotorSpeed(m_speed, motor_dir);
}

void MotorSpeed(int m_speed, bool dir){
    int pwm = m_speed % 1024;
    motor_speed = pwm;
    motor_dir = dir;
    if( dir ){
        analogWrite(PIN_MOTOR_A_IA, (1023 - pwm));
        digitalWrite(PIN_MOTOR_A_IB, HIGH);
    }else{
        analogWrite(PIN_MOTOR_A_IA, pwm);
        digitalWrite(PIN_MOTOR_A_IB, LOW);
    }
}
```

3. MotorTurn

Here a different servomotor is configured the B servomotor which is responsible for the turning of the car. MotorTurn will read a variable `turn` of which the absolute value will be calculated. If the absolute value of variable `turn` is smaller than 0 the car will go left. If this value is bigger than 0 it will turn right, and if the value is equal to 0 the car will keep going straight or is going to keep reversing, this depends on the direction of motor A. To control the motor it is the same method as in MotorSpeed. An analogue value will be sent to 1 pin of the motor more specific to the PWM (Pulse With Modulation) which determines the position of the servomotor. Then a digital value will be sent to another pin of the same motor, if this value is low the motor will reverse the rotation direction, if it is HIGH the motor will keep turning with its original rotation direction.

```
void MotroTurn( short turn ){
    force_turn = abs(turn);
    if( turn < 0 ){
        MotroTurnLeft();
    }else if( turn > 0 ){
        MotroTurnRight();
    }else if( turn == 0 ){
        MotroTurnStraight();
    }
}

void MotroTurnLeft(){
    analogWrite(PIN_MOTOR_B_IA, force_turn);
    //digitalWrite(PIN_MOTOR_B_IA, HIGH);
    digitalWrite(PIN_MOTOR_B_IB, LOW);
}

void MotroTurnRight(){
    int pwm = 1024 - force_turn;
    analogWrite(PIN_MOTOR_B_IA, pwm);
    //digitalWrite(PIN_MOTOR_B_IA, LOW);
    digitalWrite(PIN_MOTOR_B_IB, HIGH);
}

void MotroTurnStraight(){
    analogWrite(PIN_MOTOR_B_IA, 0);
    //digitalWrite(PIN_MOTOR_B_IA, LOW);
    digitalWrite(PIN_MOTOR_B_IB, LOW);
}
```

Reference List

1. <http://fritzing.org/projects/diy-wifi-rc-car-with-esp8266-and-arduino-ide>
2. <http://geek.adachsoft.com/home/article/id/27/n/DIY-WIFI-RC-car-with-ESP8266-and-Arduino-IDE>