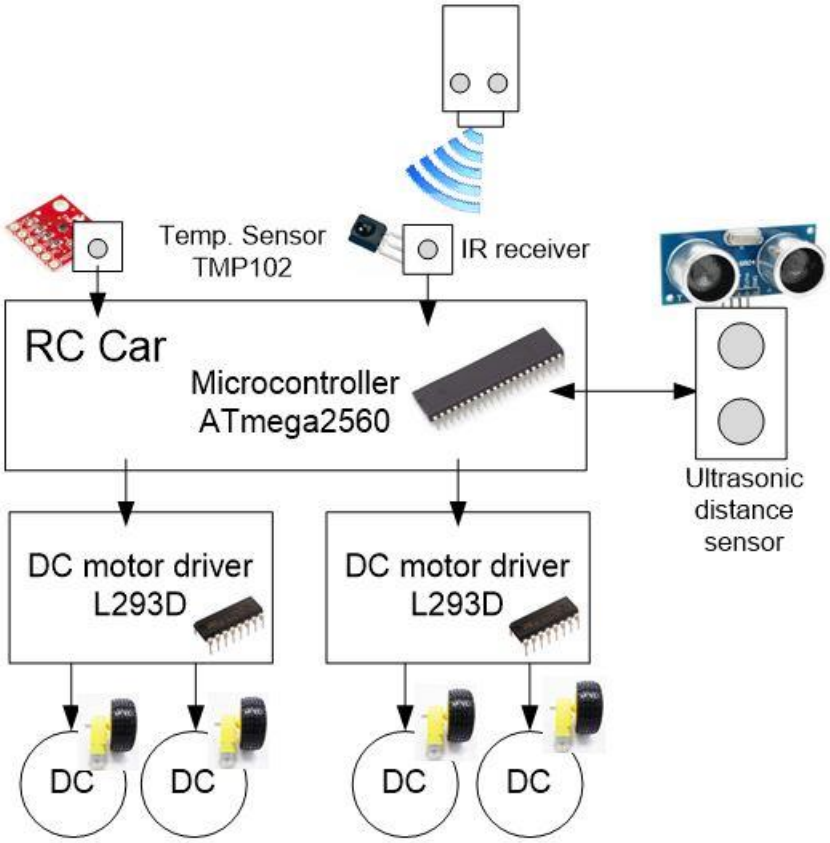
ELECTRICAL CAR

This project implements RC car using ATmega 2560 microcontroller. Car is driven by 4 DC motors regulated with motor drivers and logic given by microcontroller. 4 DC motors are controlled only by 4 wires using shift register.

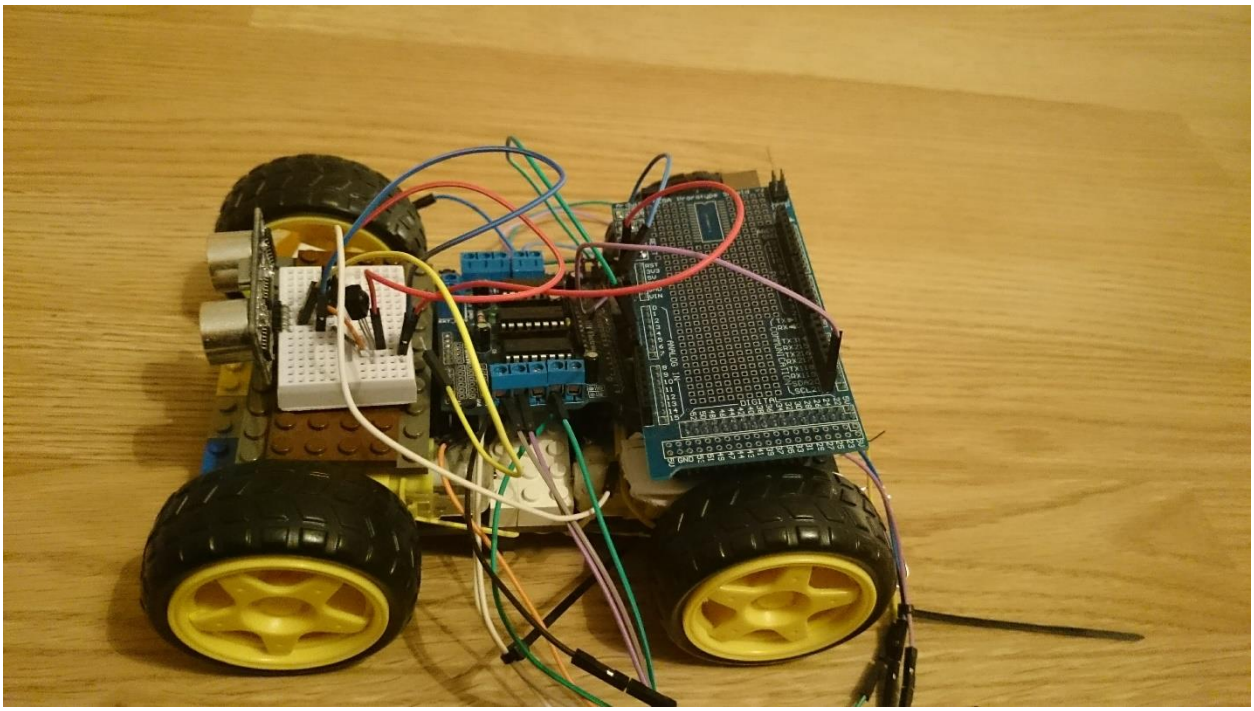


Simplified electrical car model

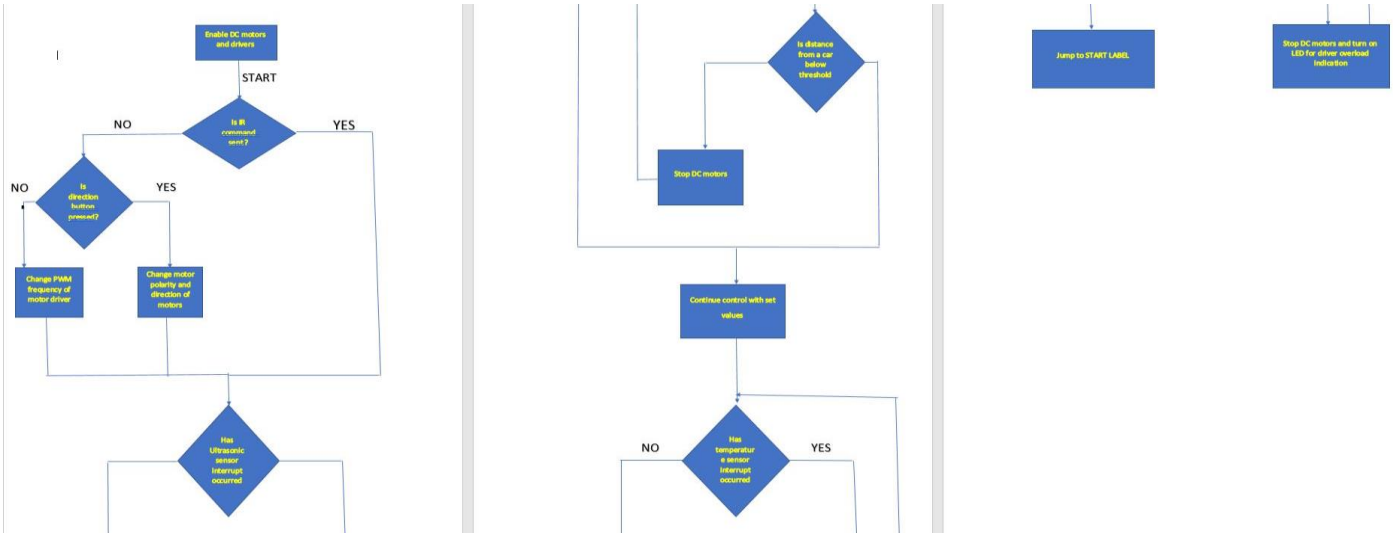
Speed of DC motors is acquired by PWM using microcontroller and motor drives. Motor drives are supplied with 7 volts which are regulated by LM317 Voltage Regulator. IR remote control is used to control whole microcontroller, where both speed and direction is regulated.

Ultrasonic sensor with appropriate algorithm is used for avoiding collision with other objects. Temperature sensor is used to measure temperature of DC motors and interrupt is given if temperature goes above threshold and all motors are stopped avoiding further damage of motor drives.

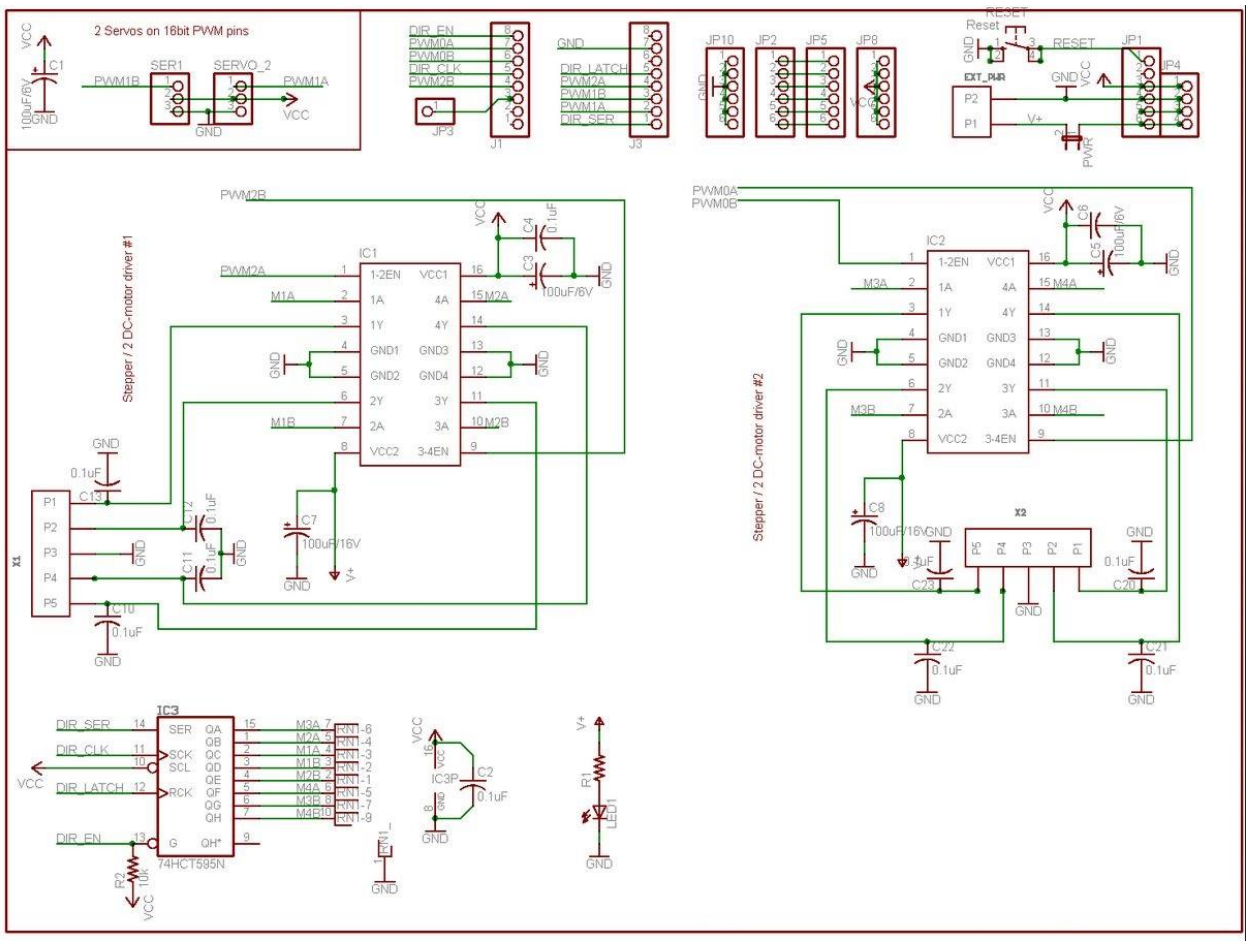
¸



Eletrical car picture



Flowchart of electrical car



Motor control schematics using two DC motor drives controlled by shift register

```
class MotorController
{
        public:
                MotorController();
                void enable(void);
                friend class DCMotor;
                void latch_tx(void);
                uint8_t TimerInitalized;
};

class DCMotor
{
        private:
                uint8_t motorNum, pwmFreq;

        public:
                DCMotor(uint8_t motorNum, uint8_t freq = DC_MOTOR_PWM_RATE);
                void run(uint8_t);
                void setSpeed(uint8_t);

};
```

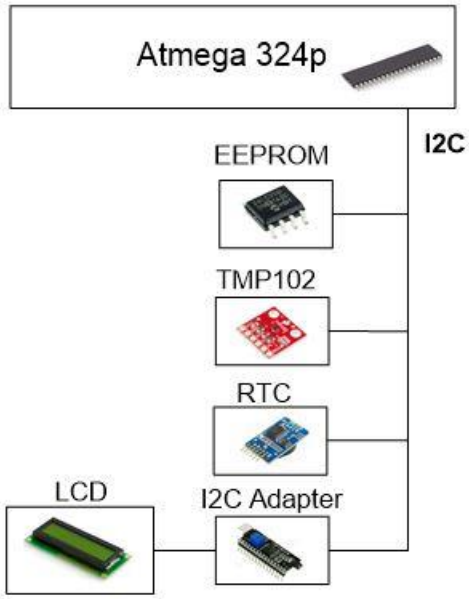
Classes for controlling DC motors written in embedded C++

ROOM TEMPERATURE REGULATION

This project implements devices needed for measuring temperature in a room. Project is implemented with I2C communication using ATmega324p microcontroller. Since I2C communication is used only two wires were used to communicate with peripheral devices.

Temperature is acquired by TMP102 temperature sensor and Real time clock is used to store date and time. All needed information is stored in EEPROM memory which is used as a data logger.

Temperature, date and time are displayed on LCD.



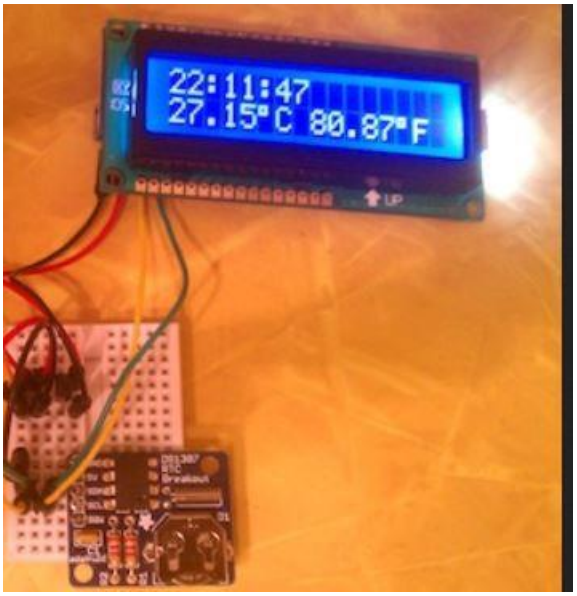
Model of room temperature measurement

To use minimum microcontroller pins for displaying data on LCD, I2C adapter is used to establish communication between LCD and microcontroller.

If user doesn't want current timer and temperature to be displayed, user can log in and check temperature for wanted date , since values are stored in EEPROM.

```
itoa(52, buffer2, 10);
USART_puts(buffer2);

i2c_start();
i2c_send(TMP102_ADDRESS_W);
i2c_send(TMP102_TEMP_REGISTER);

i2c_start();
i2c_send(TMP102_ADDRESS_R);
tempHighByte = i2c_readNoAck();
tempLowByte = i2c_readAck();
i2c_stop();


temp = 0.0625 * ( (tempHighByte << 4) | (tempLowByte >>4) );
itoa((int)temp, buffer, 10);
LCD_puts(buffer);

USART_puts(buffer);
USART_puts("TEMP:");
```

Code for communicating with temperature sensor



Date and temperature display on LCD

```
void LCD_putc(unsigned char value)
{

        //PCF8574A
        //LCD_isBusy();

        PCF8574A_PORT = (PCF8574A_PORT & 0x0F) | (value & 0xF0);
        PCF8574A_write(PCF8574A_PORT);

        PCF8574A_PORT |= 1 << RS;
        PCF8574A_PORT &= ~(1 << RW);
        PCF8574A_write(PCF8574A_PORT);

        // Rising bridge for LCD to LATCH data on its pins
        Falling_edge();

        //LCD_isBusy();

        PCF8574A_PORT = (PCF8574A_PORT & 0x0F) | (value << 4);
        PCF8574A_write(PCF8574A_PORT);

        PCF8574A_PORT |= 1 << RS;
        PCF8574A_PORT &= ~(1 << RW);
        PCF8574A_write(PCF8574A_PORT);

        // Rising bridge for LCD to LATCH data on its pins
        Falling_edge();

}
```

Function for communicating with LCD display using I2C adapter in C

Whole code can be found at: https://github.com/IvanAntunovic