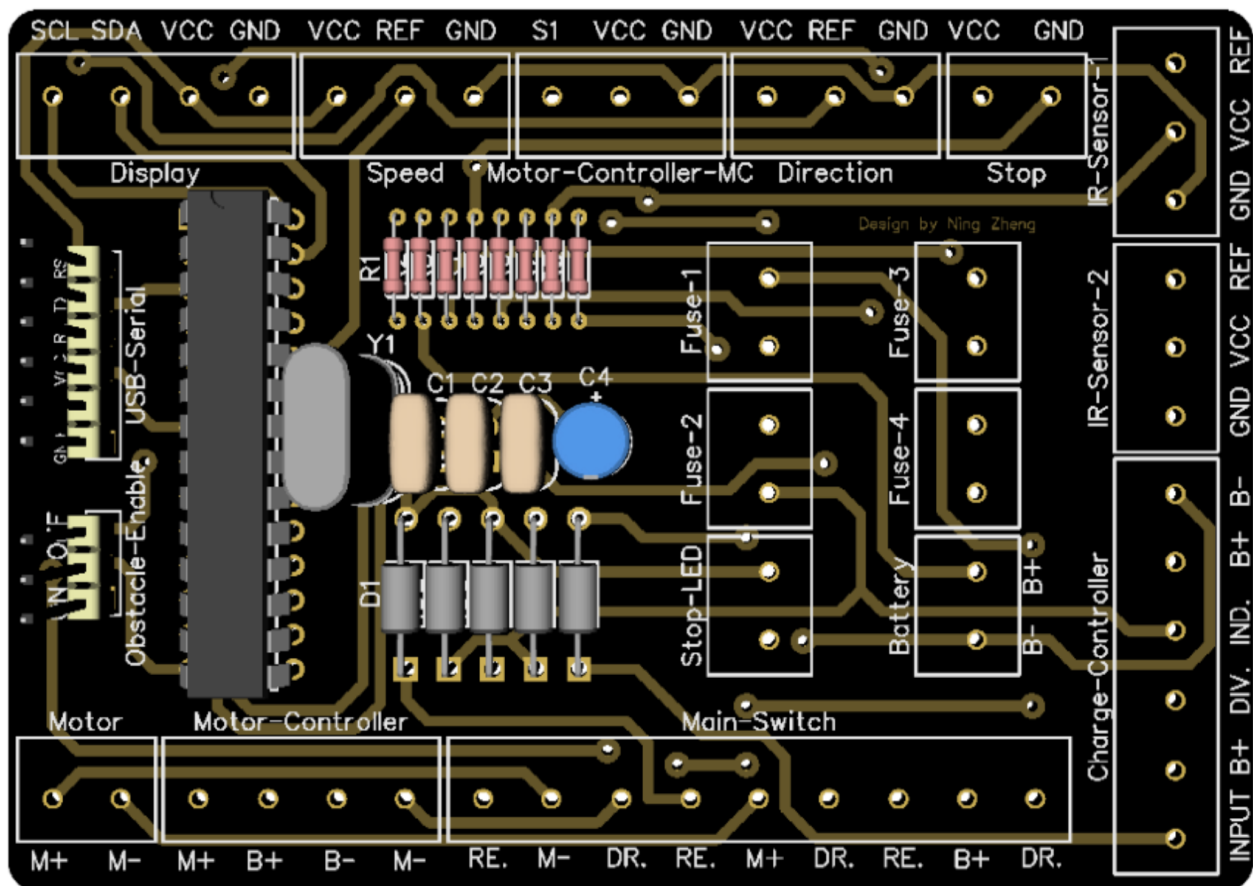


ECE 480 Application Note

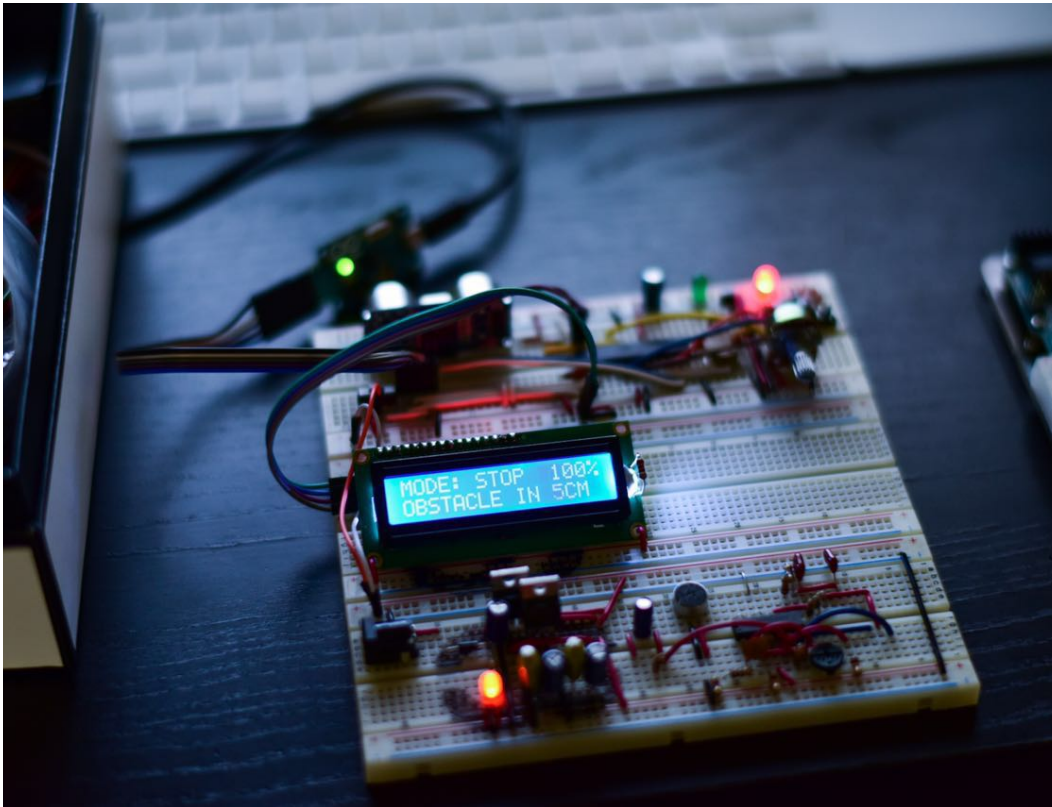
Main Control System - Design Team 9

Ning Zheng - April 22, 2016



Summary

Our project is to create a Regenerative Electric Driven Power Cart. The main purpose of this project is to create a cart that will be used in assembly plants around the world that can regenerate power without having to be plugged in. Therefore, there are two separated portions, the diving part and regeneration part. In the final design, I put those two parts into one single PCB, which can save a mount of space and make the wire connection much simple.



Final Design Prototype

Keywords: Atmega328p, IR sensor, UART communication, PCB, DipTrace

Introduction

Considering of different type of motor control, we need to choose a proper micro controller to control the behavior of the motor. There are wide variety of options for choosing a micro controller, we choose Atmega328p as our main control unit because it provides enough GPIO pins, and the response is pretty much faster comparing with the raspberry pi. And also, the Atmega328p consumes a little power which is kind to the battery powered system. We take the advantages of Atmega328p chip and built our own design.

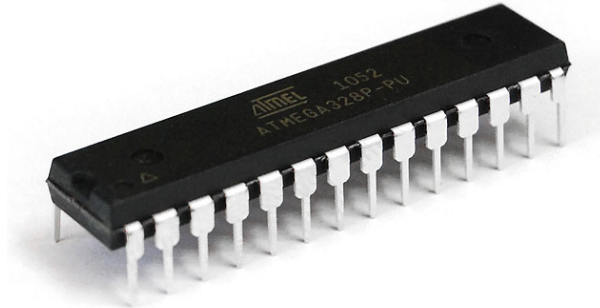


Figure 1. Atmega328p Micro Controller

Objective

The objective of this application note is to describe and outline the process of building the prototype breadboard with Atmega328p Micro Controller and developing the schematic and PCB by using DipTrace.

Procedure

1. Knowing the Atmega328p GPIO Pin configuration.

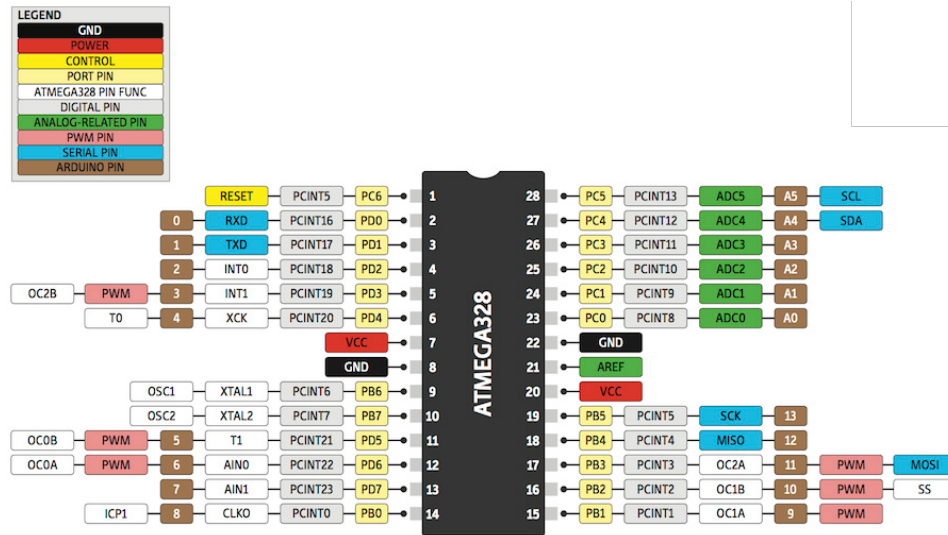


Figure 2. Atmega328p GPIO Pin

For Serial communication, we will use Pin 1, Pin 2, Pin 3, Pin 7 and Pin 8 of Atmega328p (Figure 2.) and connect it to Arduino usb2serial Light Adapter (Figure 3.).

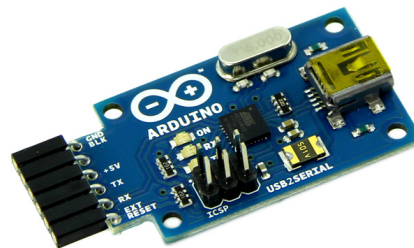


Figure 3. Arduino usb2serial Light Adapter

2. Wiring the Arduino usb2serial Light Adapter to the Atmega328p (Figure 4.).

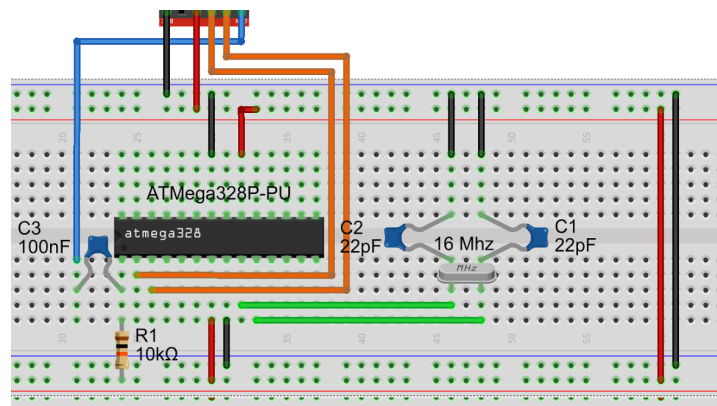


Figure 4. Arduino usb2serial Light Adapter Wiring

3. Preparing the Atmega328p for Serial Communication.

a. Tools -> Board -> Arduino/Genuino Uno (Figure 5.).

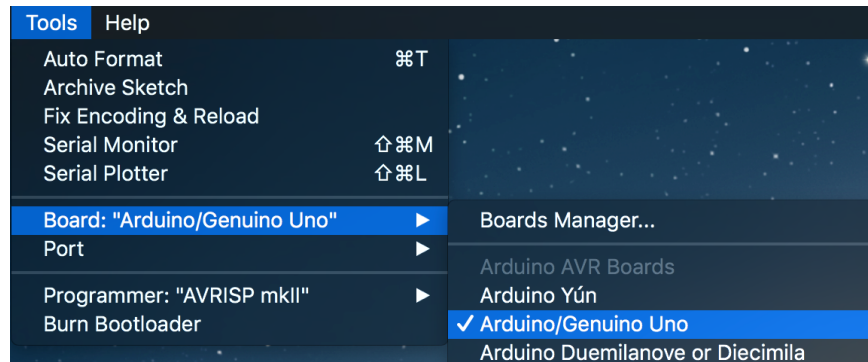


Figure 5. Arduino IDE Board Settings

b. Select the right Serial Port (Figure 6.).

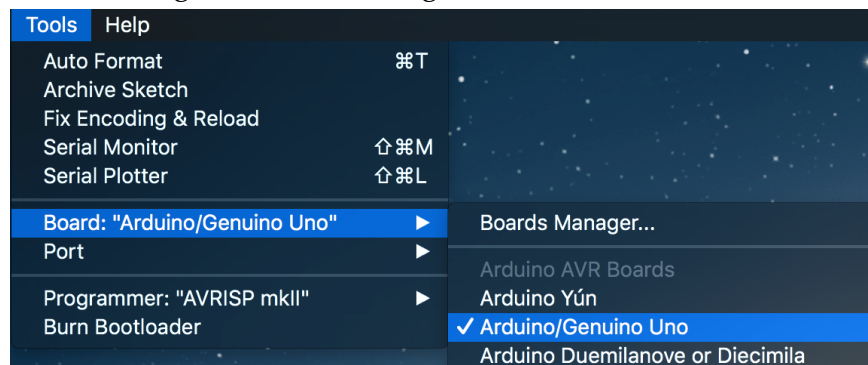


Figure 6. Arduino IDE Port Settings

c. After finishing all the set up, you can begin writing your code.

4. Interface of Arduino IDE (Figure 7.).



Figure 7. Arduino IDE Interface

- Verify. Checks your code for errors compiling it.
- Upload. Compiles your code and uploads it to the configured board.
- New. Creates a new sketch.
- Open. Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.
- Save. Saves your sketch.
- Serial Monitor. Displays serial data being sent from the Arduino or Genuino board (USB or serial board).

5. Connect IR Sensor to Atmega328p.

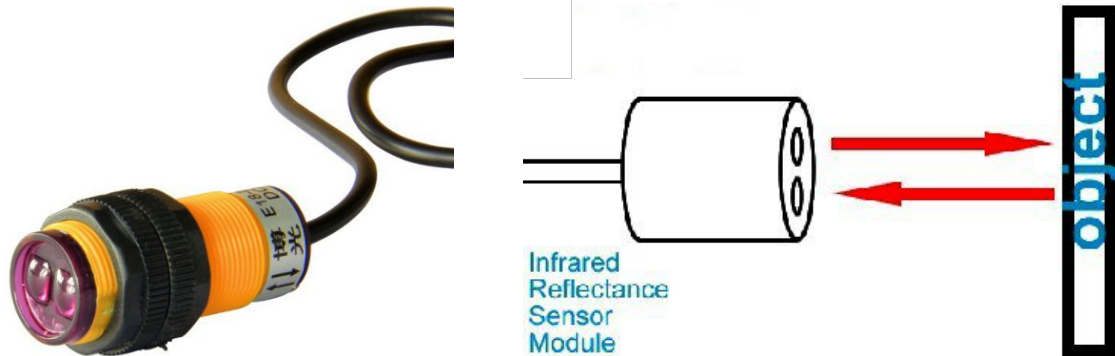


Figure 8. Geeetech Infrared proximity switch module

The Geeetech Infrared proximity switch module has three wires (VCC, GND and OUTPUT), so we need to connect the OUTPUTs to the D6 and D7 which is Pin 12 and Pin 13. And then we can use the code `digitalRead(IRSensorPin)` to show the IR sensor output.

6. Liquid Crystal Display (Figure 9.).

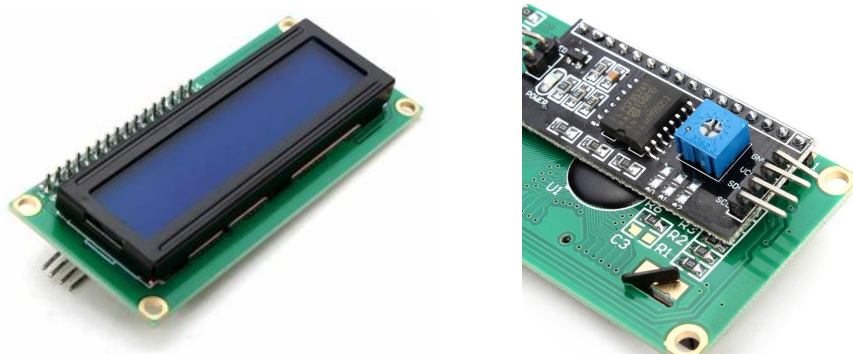


Figure 9. LCD with IIC/I2C Serial Interface Board Module

In order to get the I2C module work, we need to include a library to drive this module

```
#include <LiquidCrystal_I2C.h>
```

And then, initialize the LCD.

```
LiquidCrystal_I2C LCD(0x27, 16, 2);
```

I build a function to display content on the LCD.

```
void DisplayLCD(int line, String text) {  
  //LCD.clear();  
  LCD.setCursor(0, line);  
  if (line == 0) {  
    int space = 16 - text.length() - ReadBatteryLevel(false).length();  
    for (int i = 0; i < space; i++) {  
      text += " ";  
    }  
    text += ReadBatteryLevel(false);  
  }  
  int space = 16 - text.length();  
  for (int i = 0; i < space; i++) {  
    text += " ";  
  }  
  LCD.print(text);  
  //Serial.println(' ');  
}
```

7. Final bread board prototype.

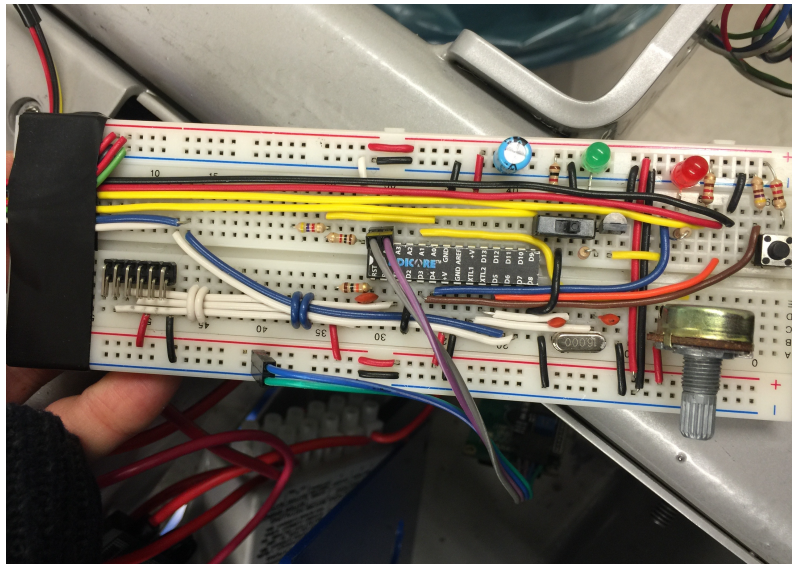


Figure 10. Final bread board prototype

8. Developing the schematic using CircuitLab.

The CircuitLab is very easy to use.

1. Select the parts we need in the left panel (Figure 11.).

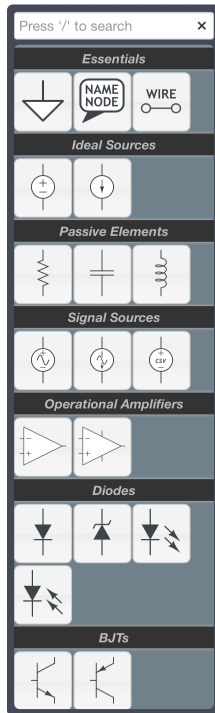


Figure 11. left panel

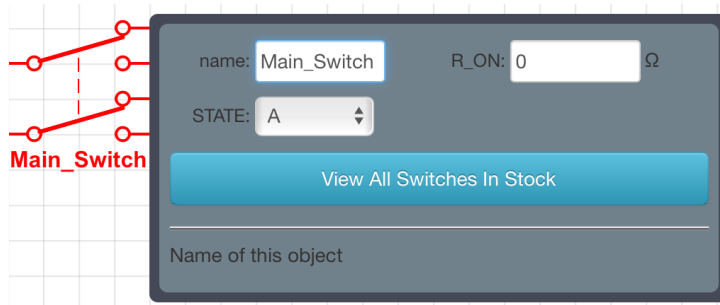


Figure 12. Attribute panel

2. Edit the attribute of the element (Figure 12.).
3. Connect the wire (Figure 13.).

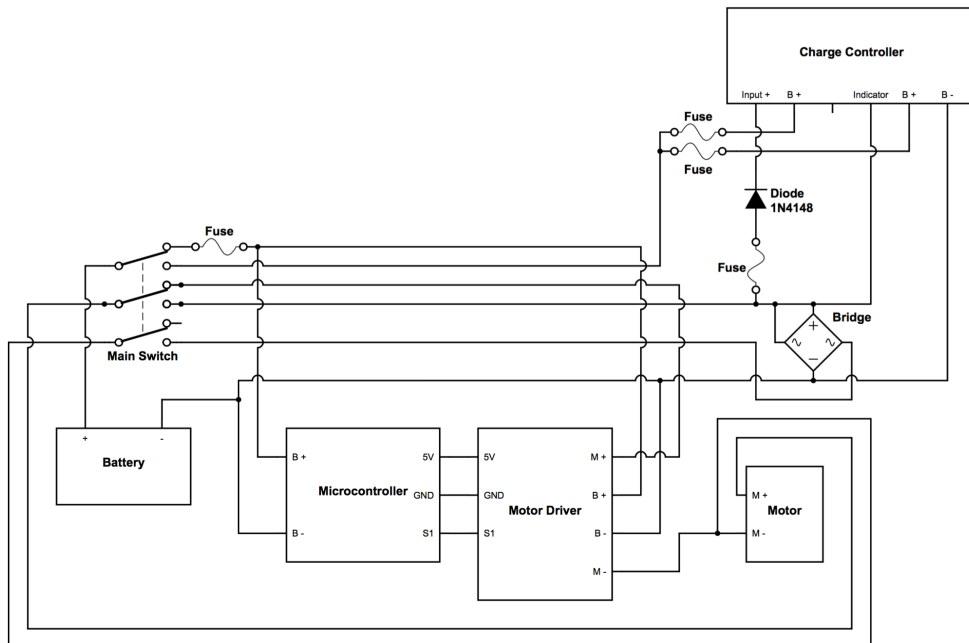
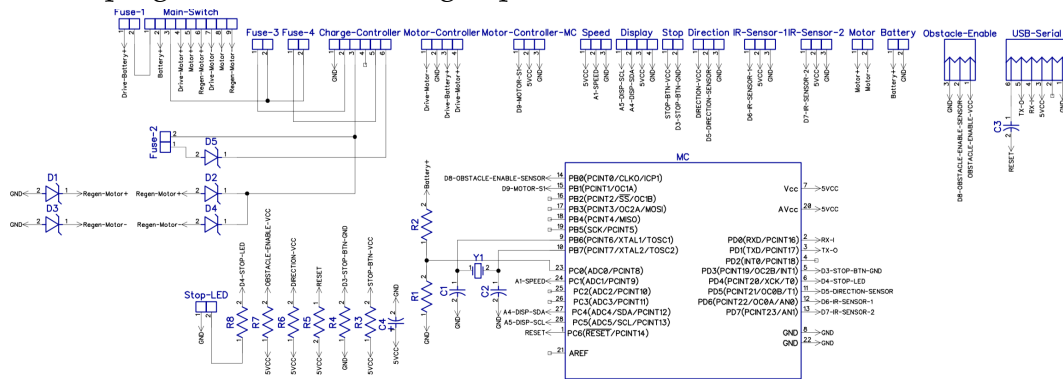
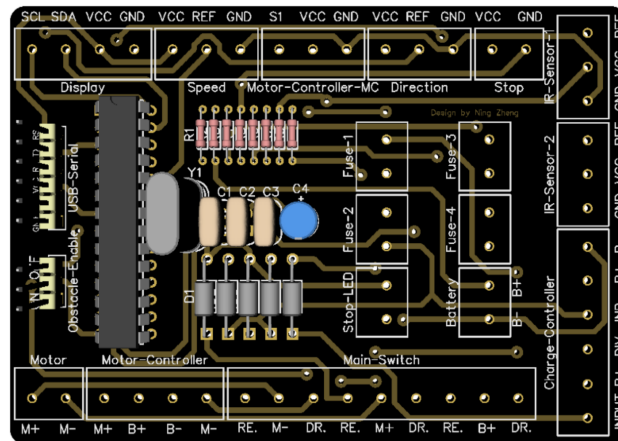


Figure 13. Final schematic

9. Developing the schematic using DipTrace.



10. Convert the schematic to PCB using DipTrace.



11.

Conclusion

This application note gives brief introduction to the Atmega328p including its pinout and wire setup. Also, it demonstrates the steps of setting up the Arduino Software (IDE) and programming with IDE. It gives examples of using IR sensor and LCD. Atmega328p is widely used not only in prototype but also in production, the Arduino UNO and Nano are all base on Atmega328p micro controller. And this application notes also gives a brief introduction to the PCB design including schematic design and PCB route.