Schematics and PCB

One of the requirements for this project is that each team makes his own circuit board. Making a circuit board yourself can be risky if you don't have any experience. That's why we have decided to make a plan A (self-made PCB) and a plan B (based on an Arduino board). We had the idea to make two boards: a board with the microcontroller, it's peripheral components and the H bridge and a board with the sensors only. If there would be a short-circuit or other failure on the self-made PCB, then the sensors PCB isn't lost and can be connected to the Arduino.

Sensor array

In a previous post, it was mentioned that the robot's sensors will consist of IR LEDs and IR phototransistors. Figure 1 shows a part of the circuit diagram of the sensor array.

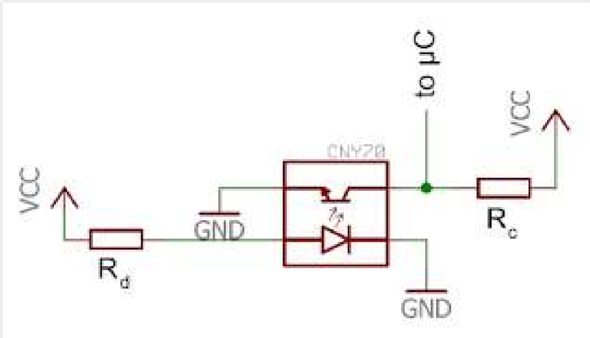

Figure 1: Common-emitter circuit for CNY70.

R_d is a current limiting resistor for the IR LED. R_c is the resistor that is used to make a voltage divider. The collector terminal of the phototransistor is connected to an analog input pin of the microcontroller. This common-emitter circuit will deliver a voltage that is proportional to the amount of light that the sensor senses (= active mode). It is the purpose to read a high voltage when the sensor senses a black surface and a low voltage when the sensors senses a white surface.

Figure 2 shows the PCB, drawn in Cadsoft's Eagle PCB. It features 6 CNY70, which are mirrored on the drawings because they will be placed on the bottom-side of the board (CNY70 are through-hole). The resistors are SMD 1206. We hope these will be easy to solder onto the board.

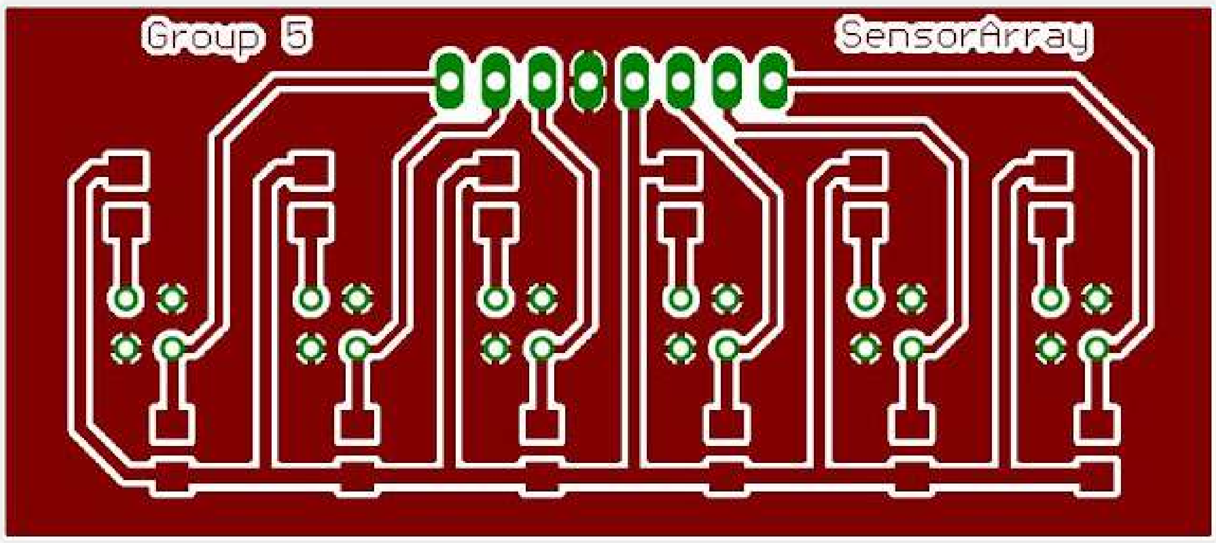

Figure 2: Sensor array PCB made with Eagle.

IR LED

An LED needs a current-limiting resistor. Using the CNY70 datasheet, we made some calculations for a forward current of 20 mA (= normal value) and 50 mA (= maximum value).

$$R_d = \frac{U - U_f}{I_f} = \frac{5 - 1{,}25}{0{,}05} = 75\ \Omega \qquad\qquad R_d = \frac{U - U_f}{I_f} = \frac{5 - 1{,}1}{0{,}02} = 195\ \Omega$$

We decided to use resistors of 150 Ω because this is a rather standard value for current-limiting resistors. The forward current trough an LED will then lie between 20 mA and 50 mA.

IR phototransistor

According to multiple application notes from electronics manufacturers, R_c is typically less than 5 kΩ and the exact value of R_c can only be found by testing. So, we decided to test one sensor with R_d = 150 Ω and R_c = 5 kΩ (10 kΩ || 10 kΩ). The collector terminal of the sensor was connected to an Arduino Uno, which read the analog signal and sent it serially to the computer. The measured values were shown on Arduino's IDE Serial Monitor. The black/white surface was simulated by a sheet of paper with a printed black line.

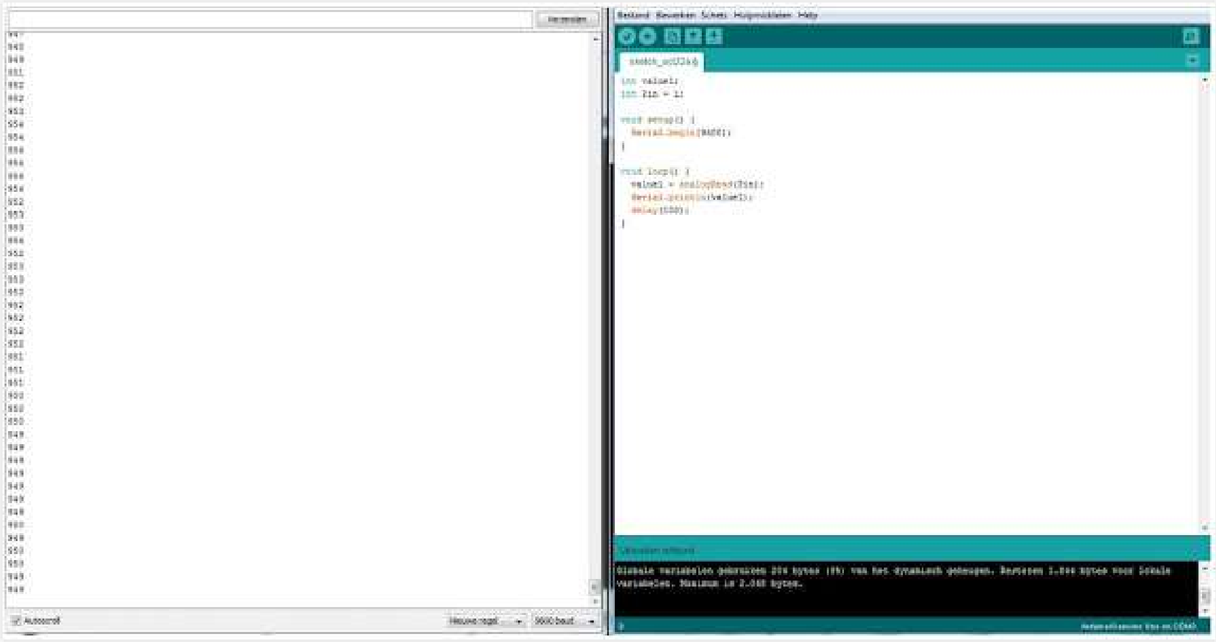

Figure 3: Left: Serial Monitor. Right: Simple Arduino program to read analog values.

We found that the values varied between 49 and 954 (on a scale from 0 to 1023). Since 5 kΩ isn't a standard resistor value, we will use a smaller value: 4,7 kΩ.

Main board

Figure 4 shows the schematic of the main PCB. There are four main groups:

- power supply and voltage regulator
- H bridge
- microcontroller and it's peripheral components
- bluetooth communication

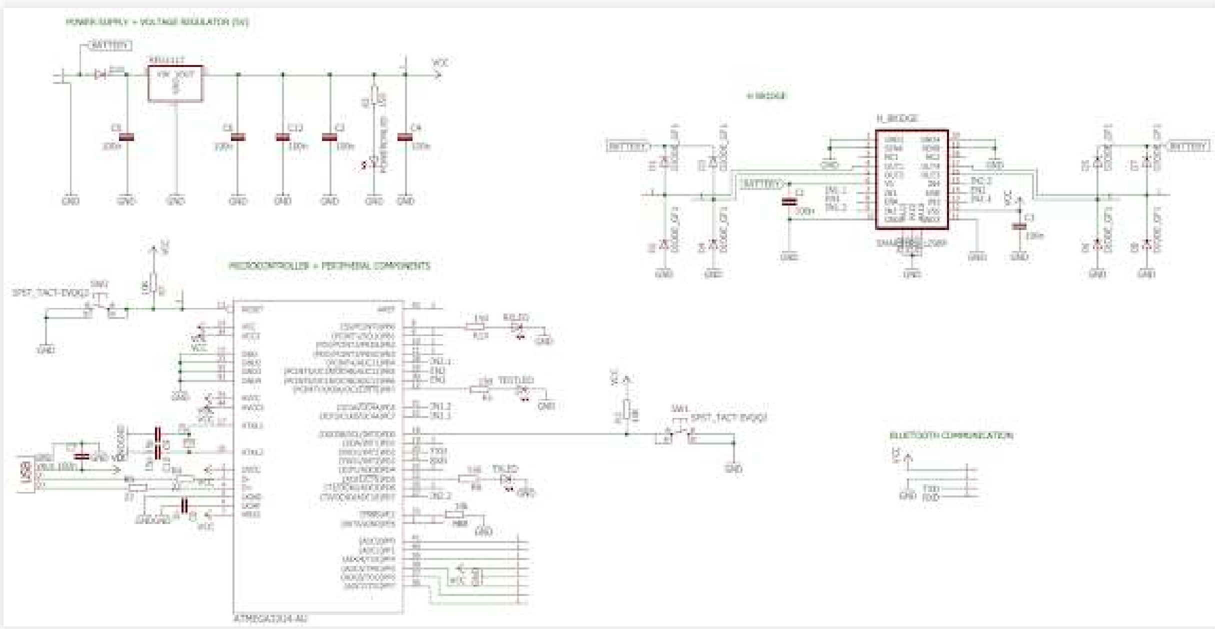

Figure 4: Schematic of the main PCB.

Power supply and voltage regulator

The power supply consists of 6 AA batteries, providing 9 V and has a total capacity of 3300 mAh. The 'BATTERY' tag leads to the H bridge, whereas the VBAT+ terminal goes further to the 5 V voltage regulator. A diode is placed to make sure that the current does not change direction if the batteries are running low. Pin 2 of the voltage regulator provides 5 V, which is labeled as 'Vcc' in the diagram. The 100 nF capacitors filter out noise and an LED with 150 Ω resistor shows whether the power is on or off.

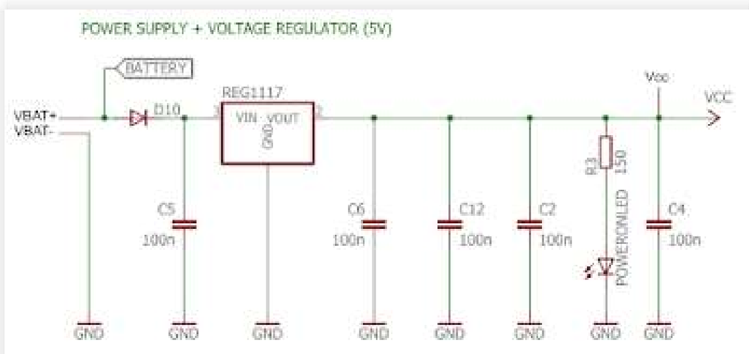

Figure 5: Power supply and 5 V voltage regulator.

H bridge

The L298P can control two motors in both directions. The direction can be chosen with the pins IN1.1, IN1.2, IN2.1 and IN2.2. We want to adjust the motor speed using PWM. This can be done by modulating the ENA and ENB signals that come from the µC. The manufacturer of the L298P advises to use Schottky diodes (symbol on drawing is wrong) as freewheel diodes. We chose for Schottky diodes with 3 A forward current in a DO-214AC package because these diodes are faster than regular ones. The diodes need to be fast because of the high-frequency PWM signals that control the motors. On the VS and VSS pins, there are 100 nF capacitors placed to filter noise.

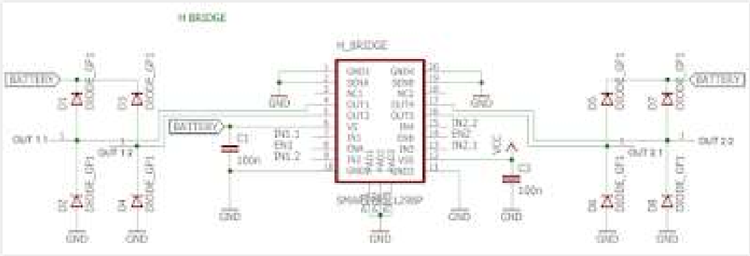

Figure 6: H bridge.

Microcontroller and peripheral components

Like we've already said, we want to use the ATmega32u4 microcontroller. The µC works at a clock frequency of 16 MHz. This crystal is connected to the XTAL1 and XTAL2 pins, together with two 15 pF capacitors. A mini USB connector is connected to the D+ and D- pins with two 22 Ω resistors (see datasheet of µC). The µC can be reset with a push button (with 10 kΩ pull-up resistor). The RESET, MISO, MOSI, VCC, GND and SCLK pins are needed to burn the bootloader onto the µC. That's why there are pin headers connected to the pins of the microcontroller. The analog input ports (PF0 to PF7) are also connected to pin headers.

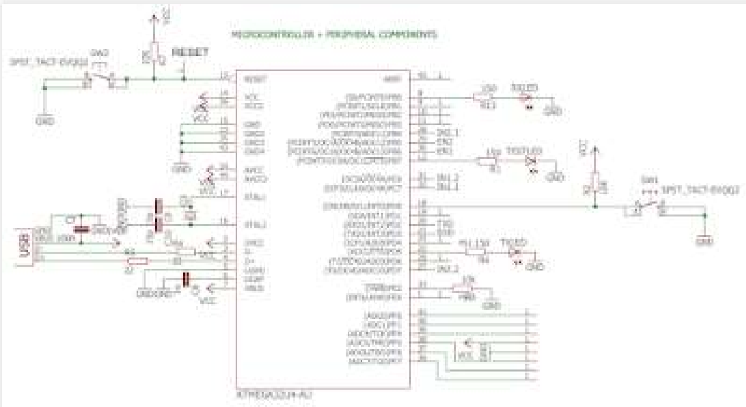

Figure 7: Microcontroller and peripheral components.

The push button on the right side of Figure 7 will serve as the start/stop button for the robot. It is useful if the microcontroller lets you know if he sends or receives serial data. This can be done by placing an LED with 150 Ω resistor on the PB0 and PD5 pin. The Test LED can be used to do the popular 'blink' test.

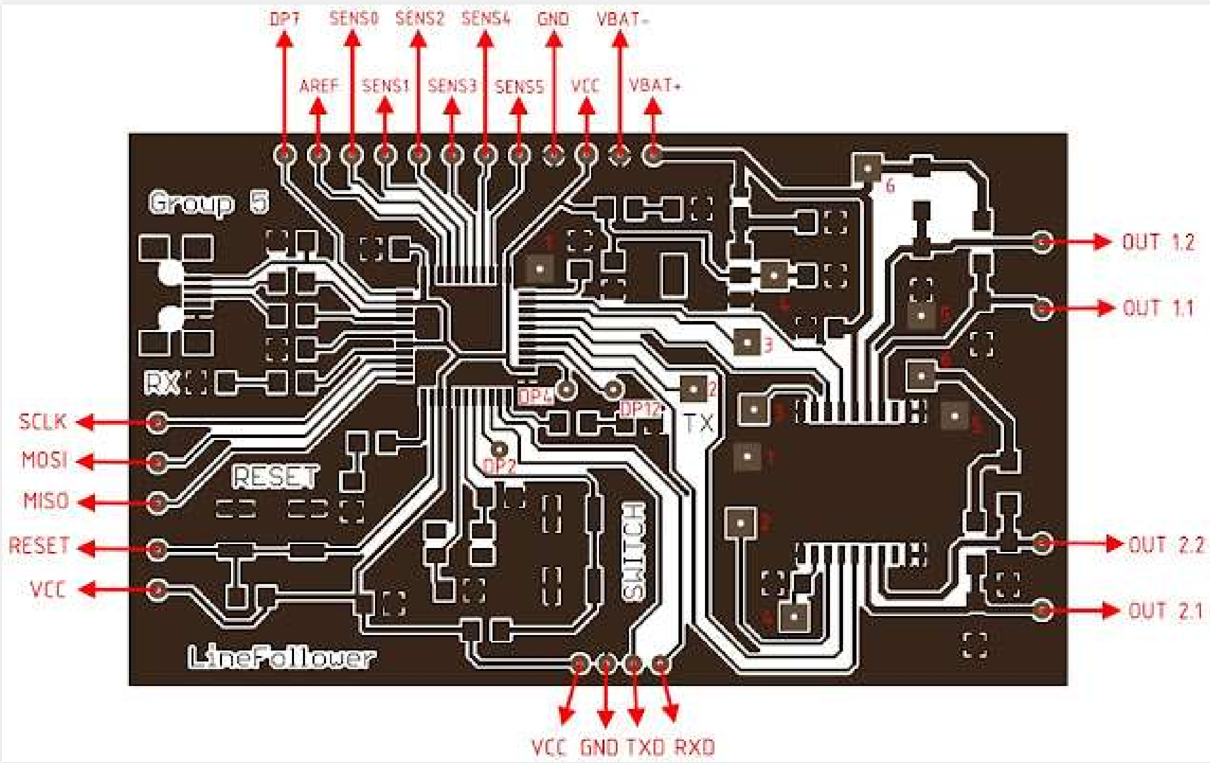

Figure 8: Etch of the main PCB.

Bluetooth communication

We will use a JY-MCU bluetooth module for wireless communication with a laptop or smartphone. The JY-MCU module has 4 pins: Vcc, Ground, TXD and RXD. The schematic in Figure 8 shows 4 pins that need to be connected to the module. The TXD label is connected to the RXD pin on the µC and vice versa.

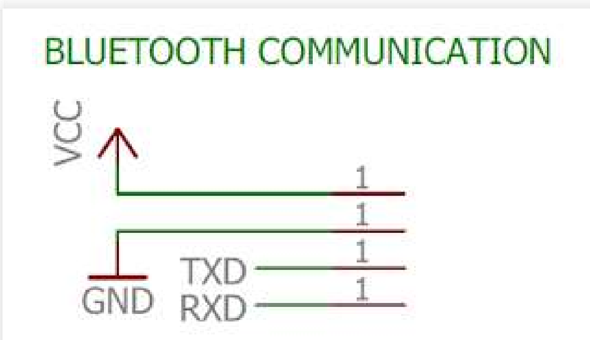

Figure 9: Bluetooth communication.

13th December 2015 gepost door Anonymous