Arduino 4WD Car - Assembly and Code - With Optional Infrared or Bluetooth Control and Autonomous Driving Capability

by DerGlorreiche

Assembly instruction for a cheap (30€) chinese 4WD_KIT1 and how to improve and extend it. The control software is available as an Arduino library.

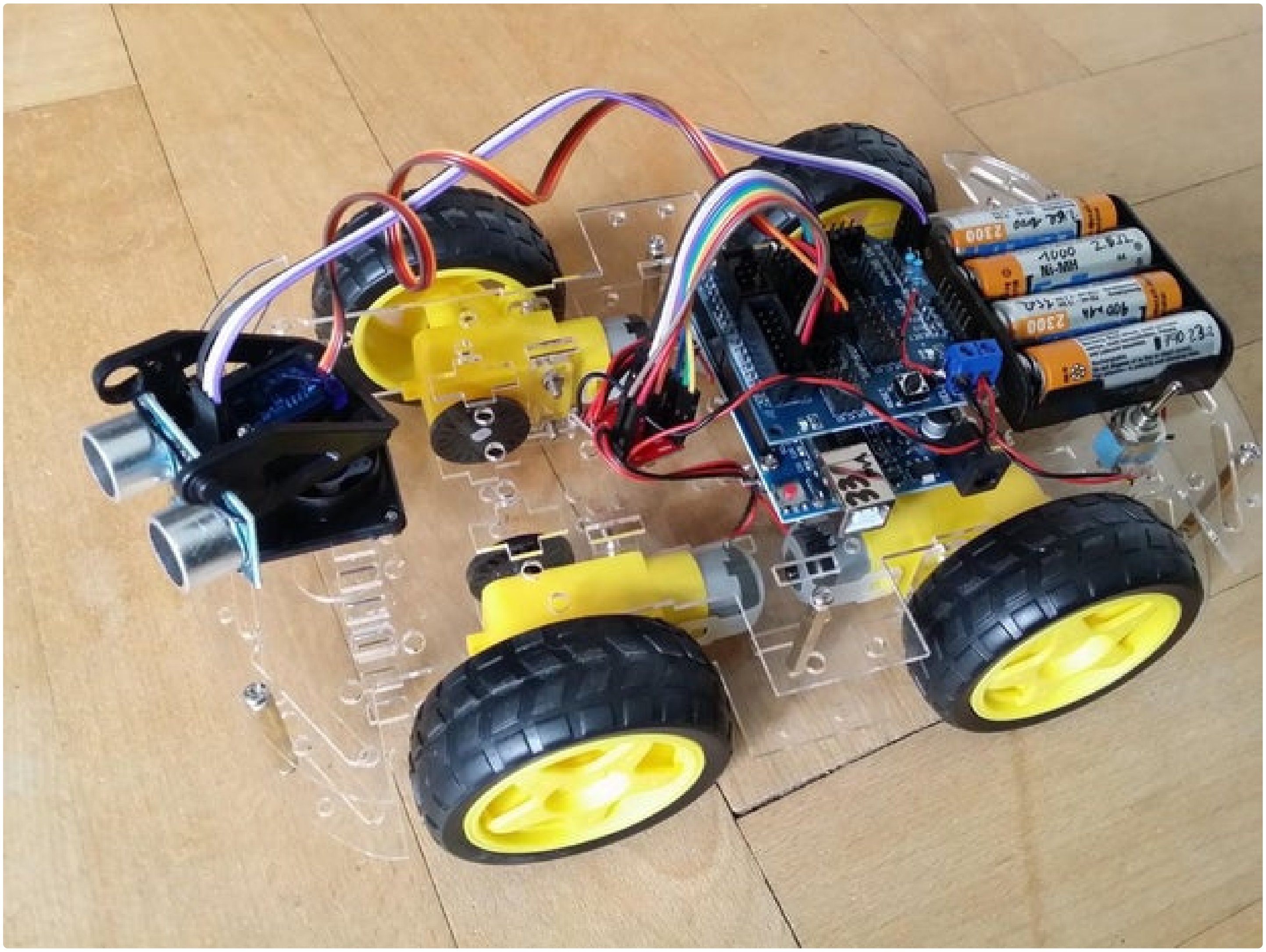
Supplies:

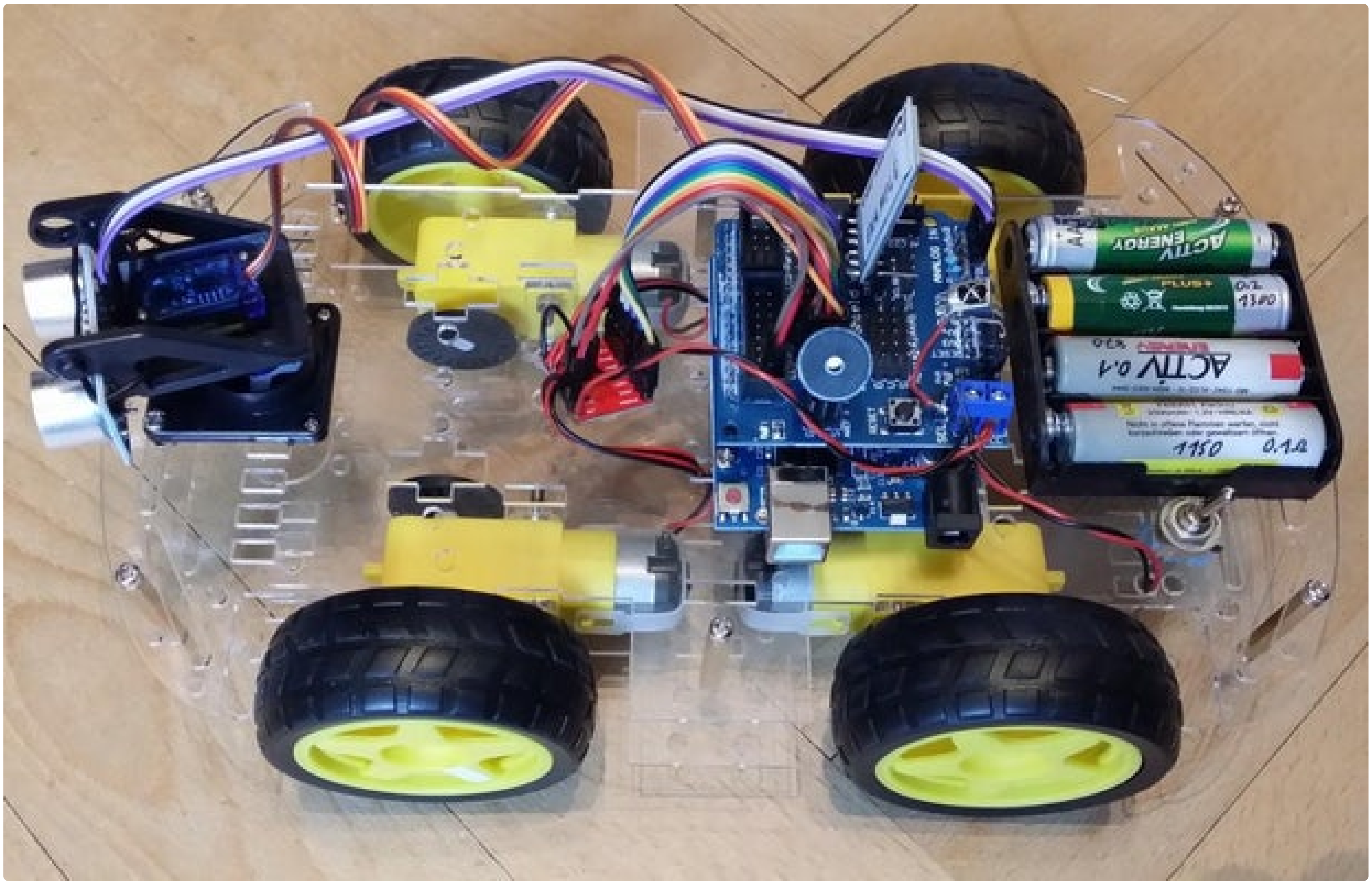
- Avoidance tracking Motor Smart Robot Car
- TB6612 Dual Motor Driver 1A (replacement for the awful inefficient LM298 driver)
- 4 wire 20 cm female-female dupont cable for HC-SR04 connection (missing in my china set)
- Toggle switch (missing in the china set)
- Two additional M3 screws and nuts to fasten the uno board
- 10 wire 12 cm female-female dupont cable for the TB6612 connection (if you do not want to use the 15 cm cable intended for the IR sensors, which we do not use here)

Optional:

- AC2208 Piezo buzzer (for audio feedback)
- 10 kOhm + 100 kOhm 1% resistor for optional undervoltage protection and supply voltage display in the Bluetooth control GUI.
- HC-05 Bluetooth modul for bluetooth control
- TL1838 IR receiver for IR control
- Female 20 pin header row to build adapters for the buzzer, the IR receiver and optional the motor power connectors and BT module.

A Phillips screwdriver, a cutter for the servo connector, a soldering iron, a 2 mm and a 5 mm drill and a few cm wires (and a tweezer) are required for assembly. A hot glue gun is helpful for fixing the HC-SR04 ultrasonic sensor at the servo rack.



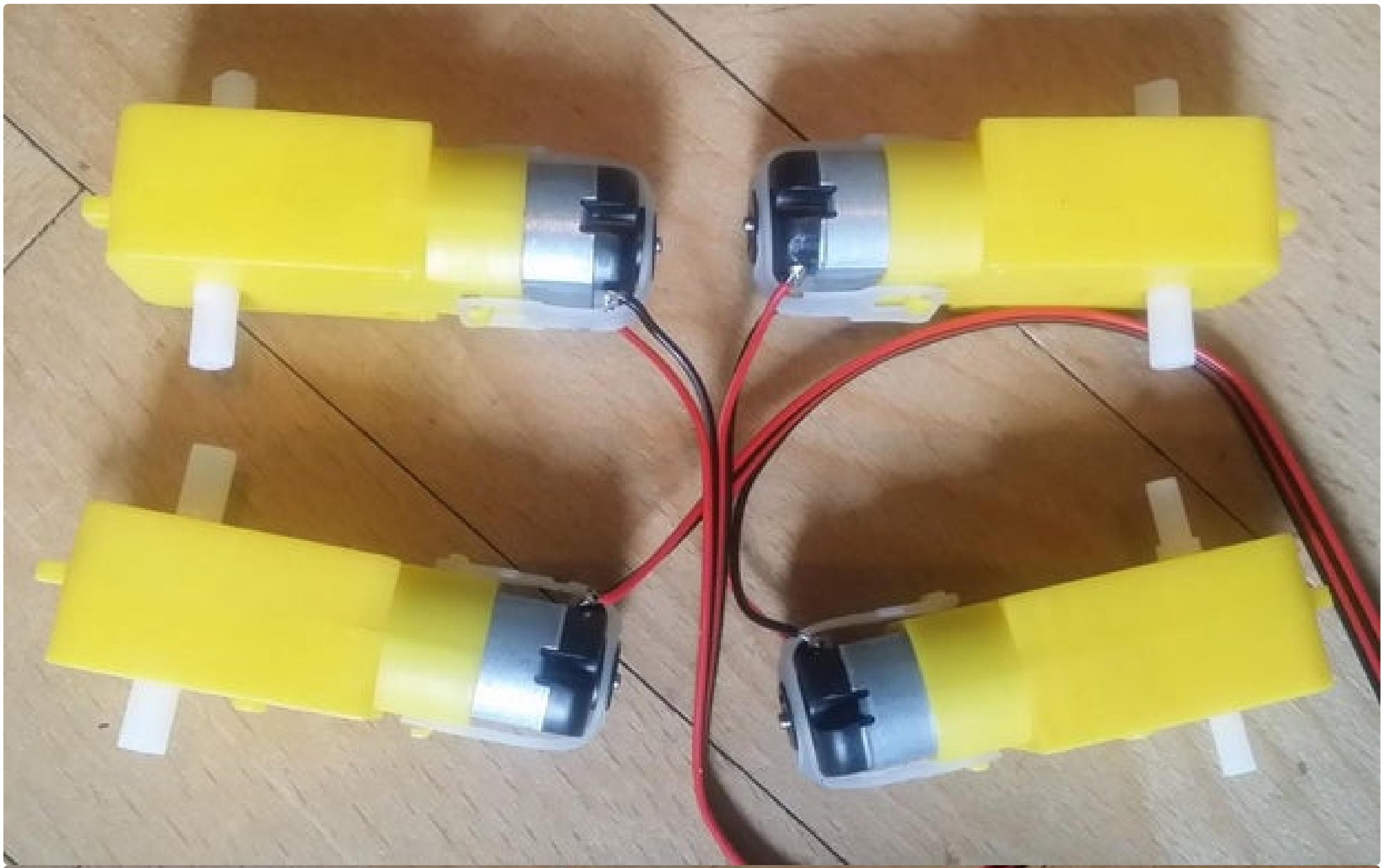


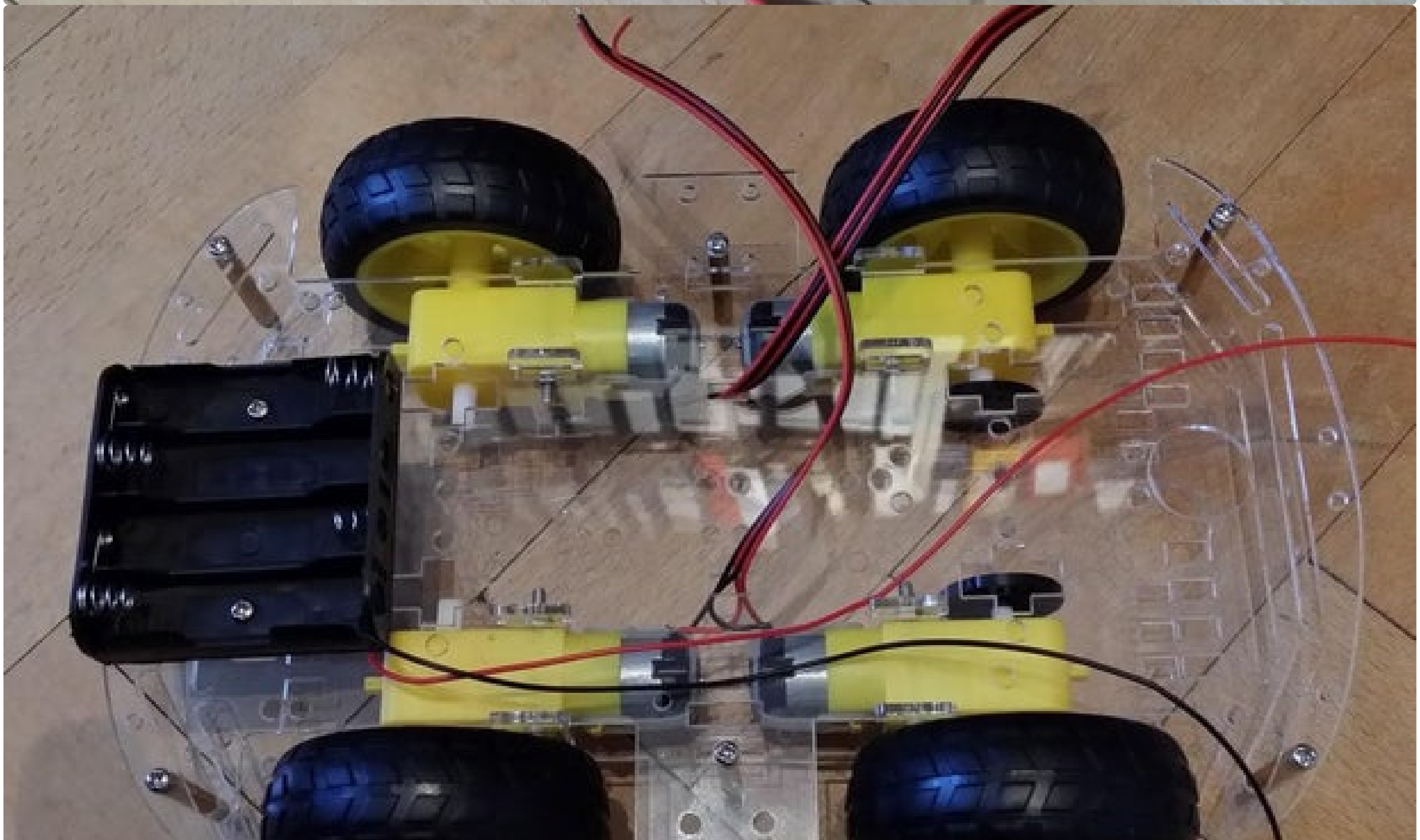
Step 1: Motor Cabling and Assembly

For a 4 WD car **all motors of one side should run at the same speed**so both motors are supplied by the same power source.

I mounted the motors with the **terminals side by side** and therefore plus and minus connections must be swapped between the front and left motor. Plus and minus must also be swapped between the left and right motors.

The **optical interruptors are useless** without slot-type photo interrupters mounted and an elaborated software support, but they look cool in a special way :-)



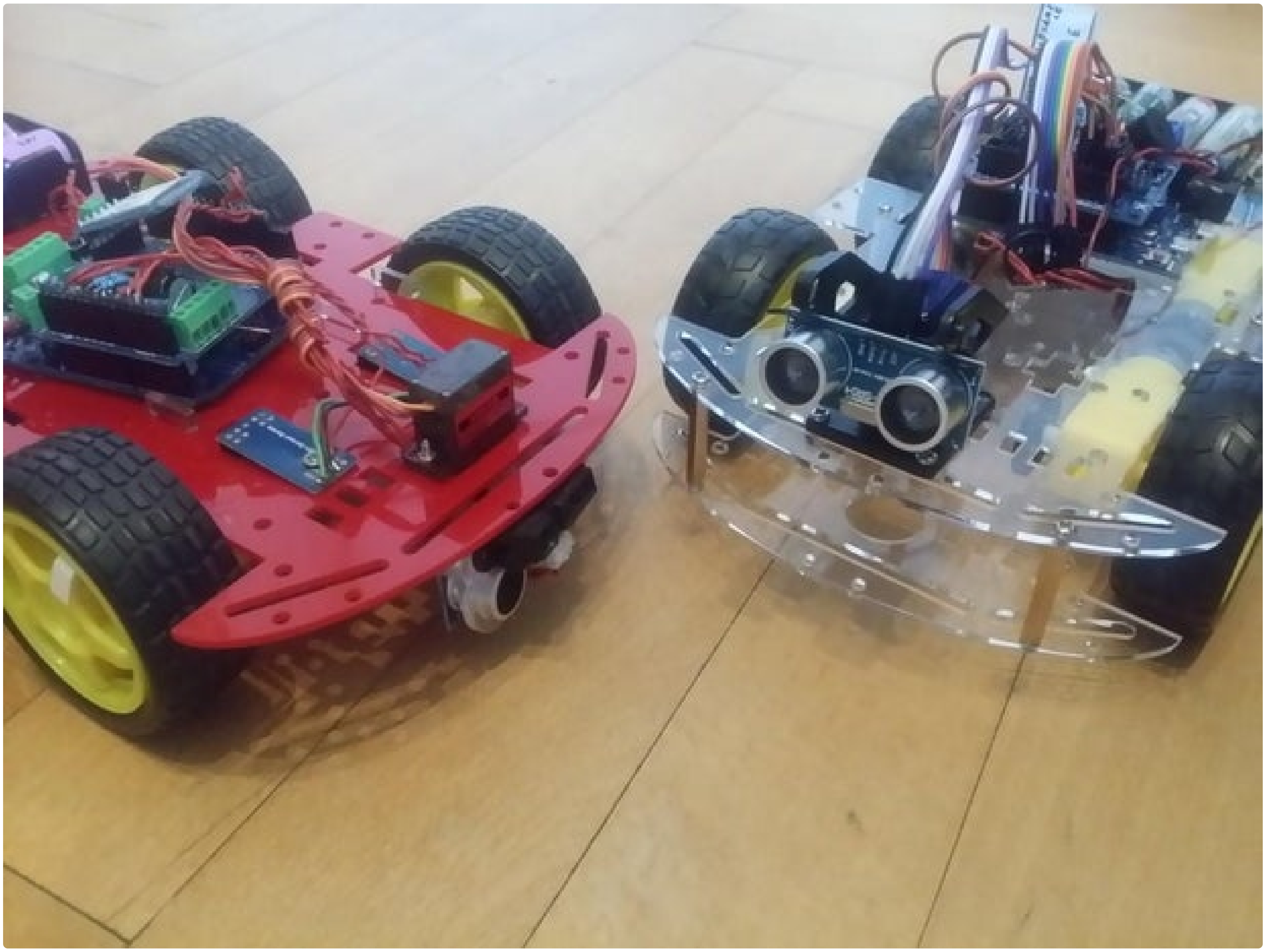


Step 2: Chassis Flavors

The chassis can be built in 2 flavors, with one or two decks. The two deck version maybe more sturdy, while the 1 deck version looks more sportive. It also allows to mount the distance sensor below the deck, which enables it to detect even small obstacles.

For the 2 deck version **the motors are mounted at the bottom deck**, and all **the other parts are mounted at the top deck**.

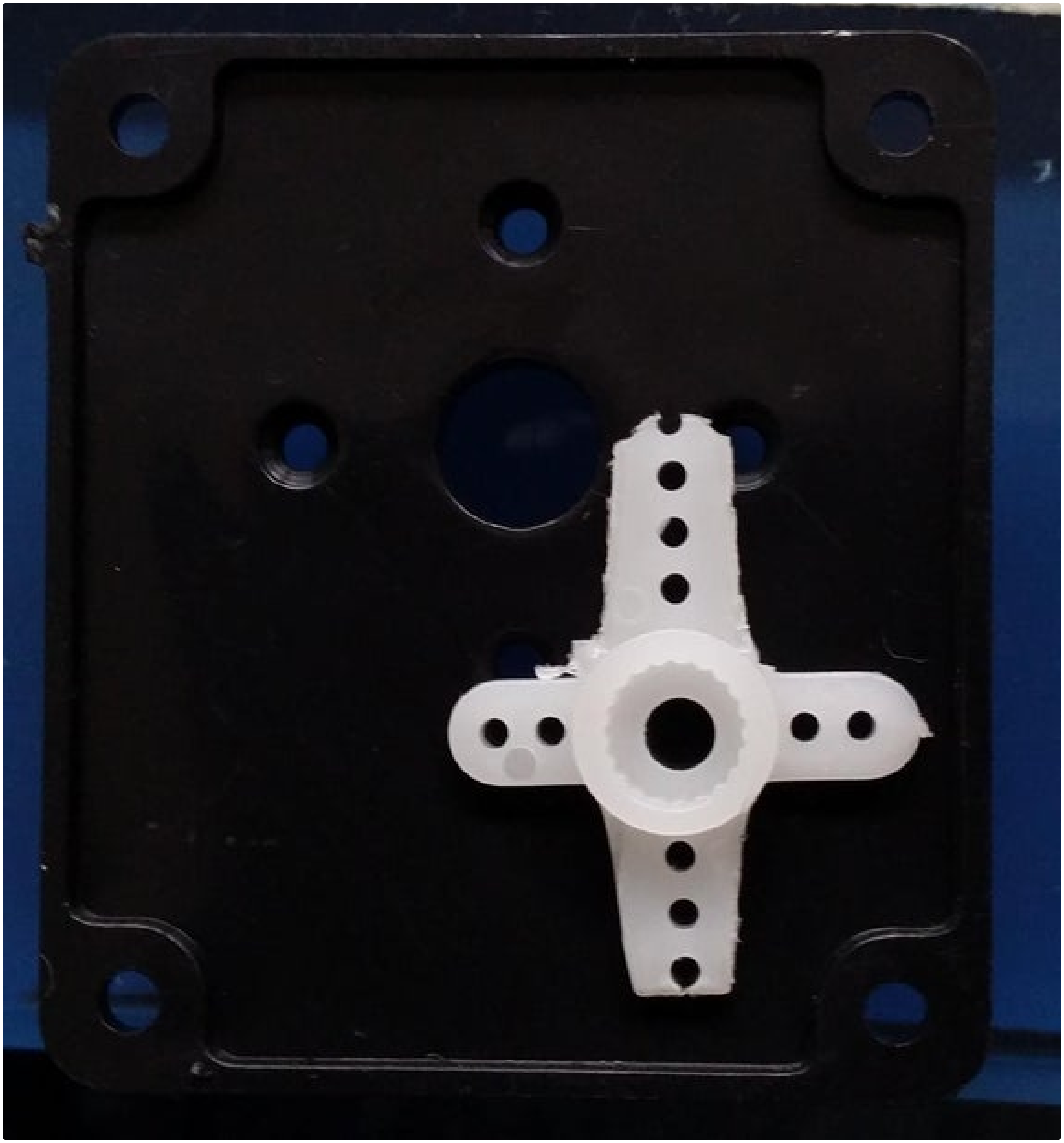
Here you find an assembly video for the 2 deck chassis.

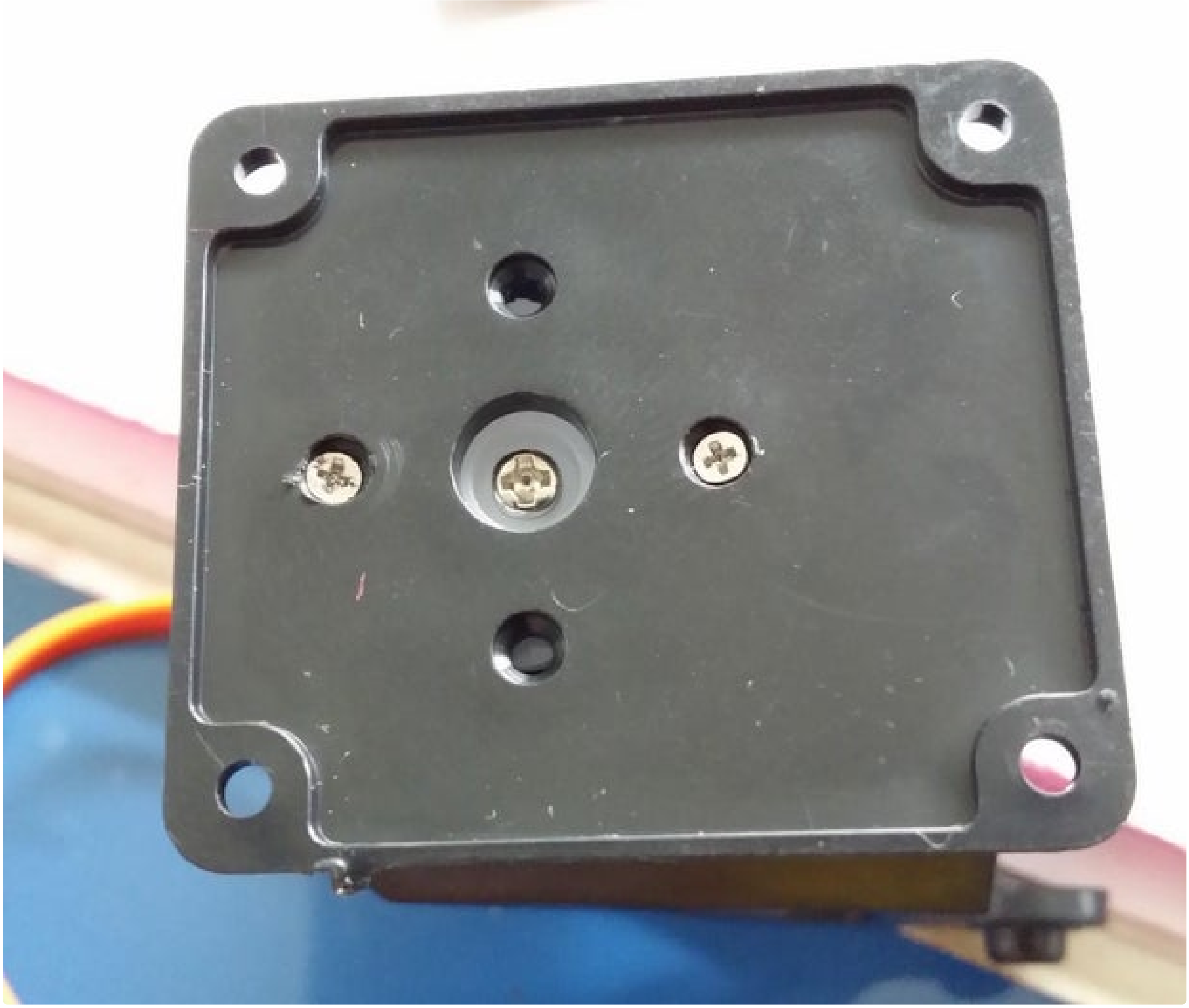


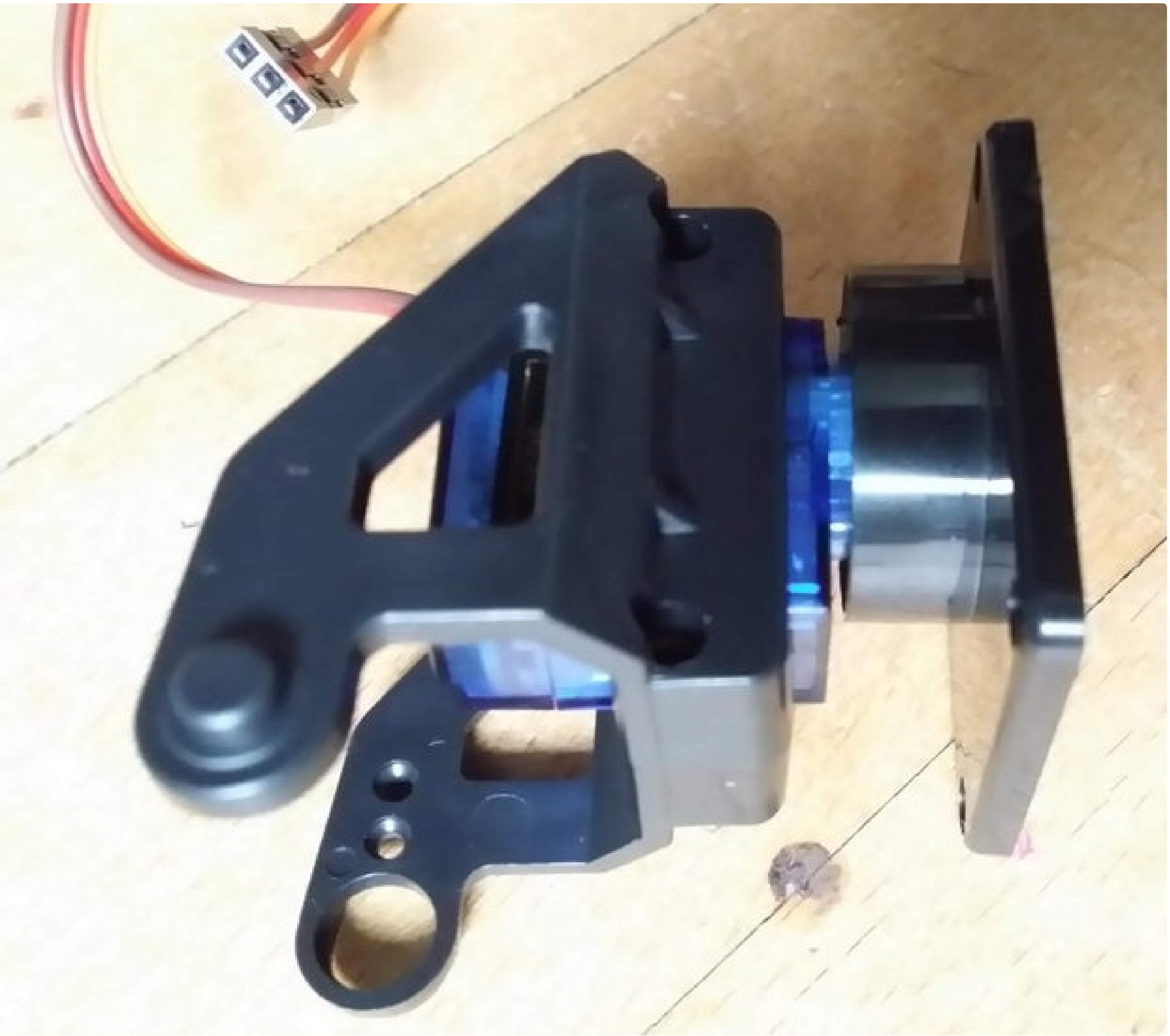
Step 3: Ultrasonic Distance Servo Mounting

Mounting the delivered servo and ultrasonic distance sensor requires manual work, since it is not supported out of the box.

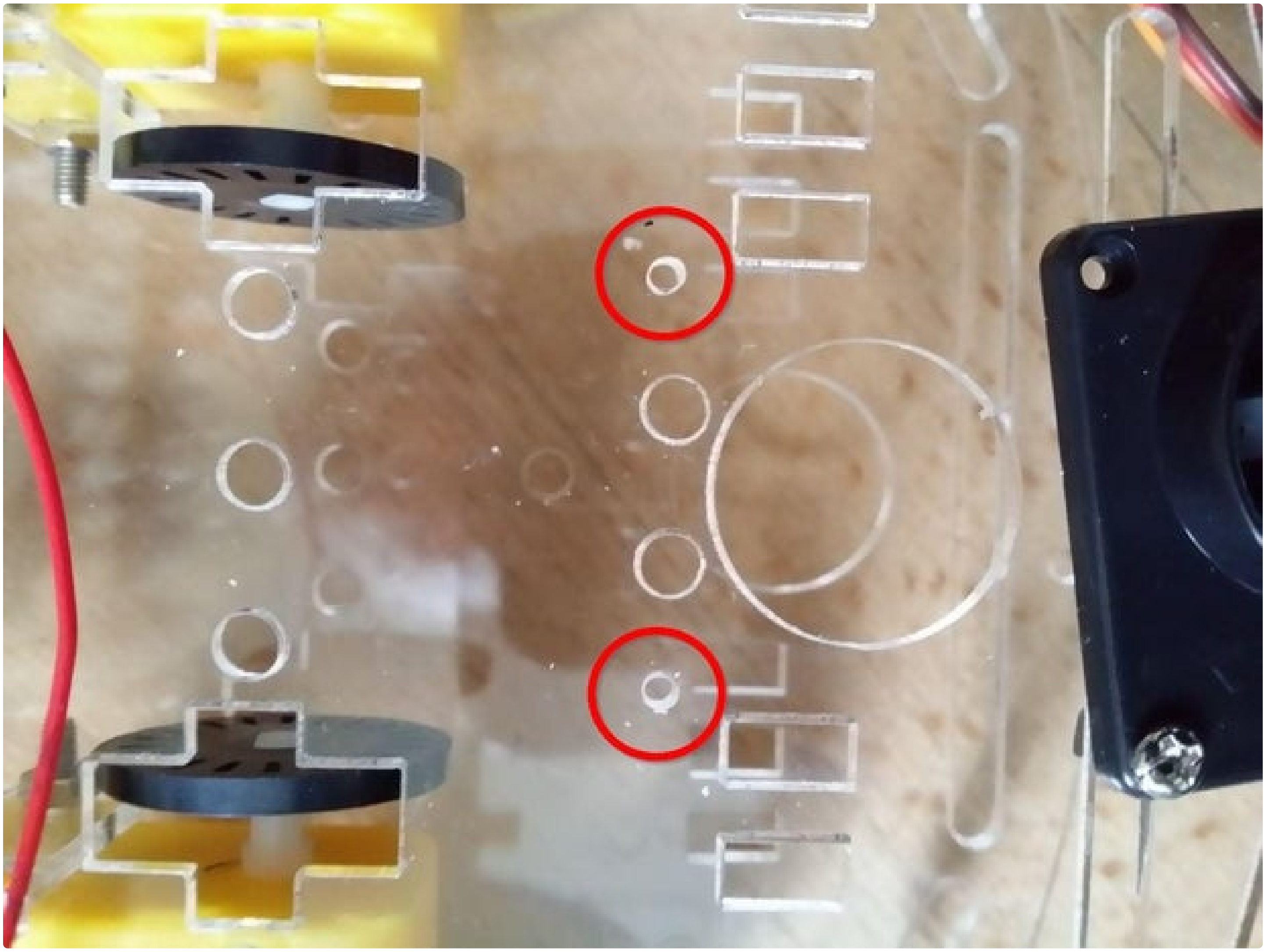
- Trim the two longer arms of the servo connector, so they have only 3 holes and their ends are narrow enough to fit in the base. Use the two 3mm countersunk self-tapping screws to attach it at the base.
- Set the servo to 90 degree by using a program or choosing the middle position between the two end stops - attaching the two-arm servo connector helps finding the right position.
- Use the panhead screw to attach the servo connector to the servo.
- Use the two 7 mm screws to connect the two halves to clamp the servo.
- Strap the ultrasonic sensor, pins up, to the mount with hot glue or twist tie.
- Widen the front holes of the base with a scissor, a tweezer or a similar tool for the M3 screws to pass through.
- Fasten the base with 2 screws in the front slot and position it in the center.
- Mark the position of the back holes, drill two 2 mm holes in the top deck and use the M2 screws of the servo set for it. Using only the 2 front screws has the risk of breaking the front slot if you accidentaly apply to much force to the sensor / servo structure.

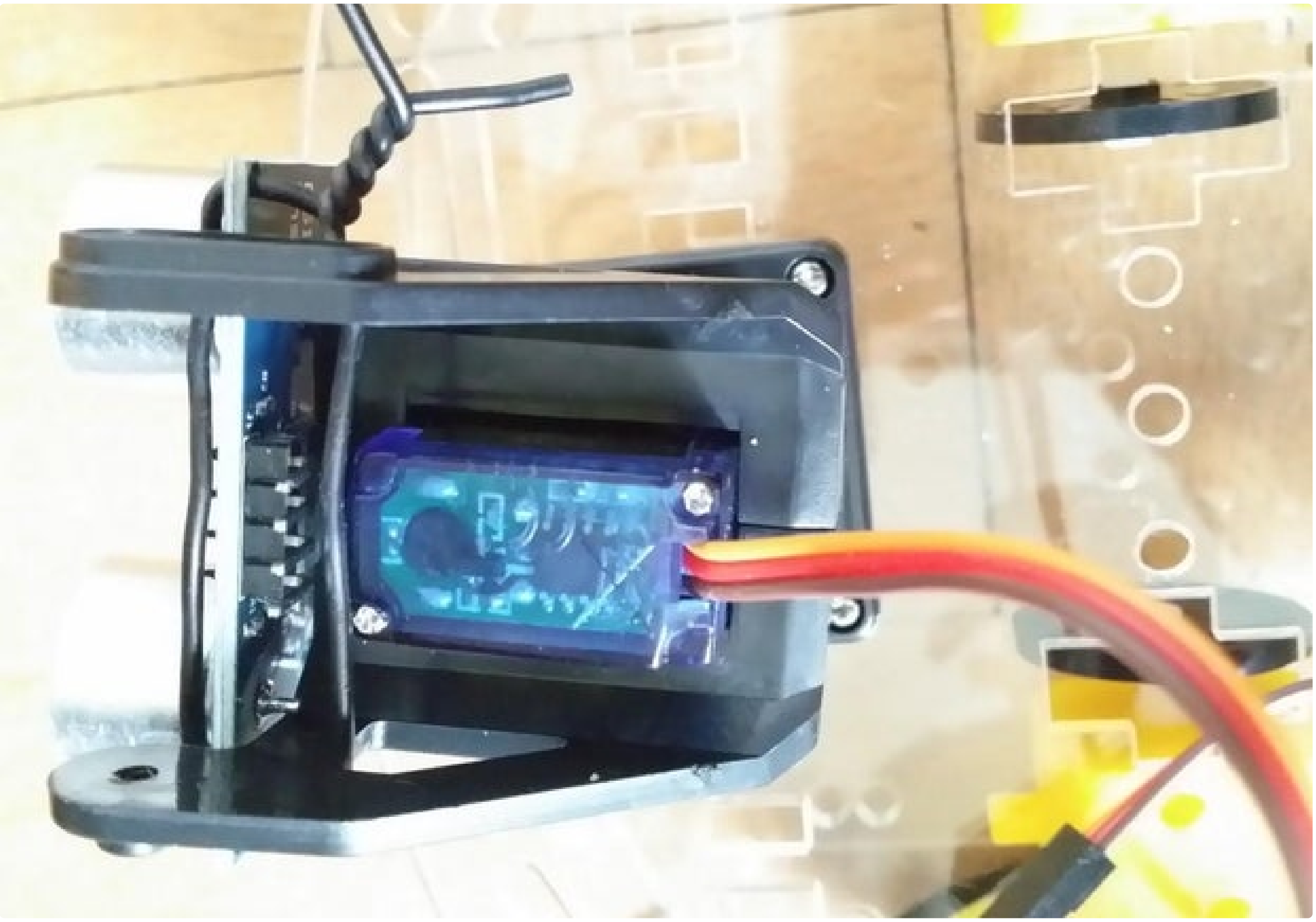


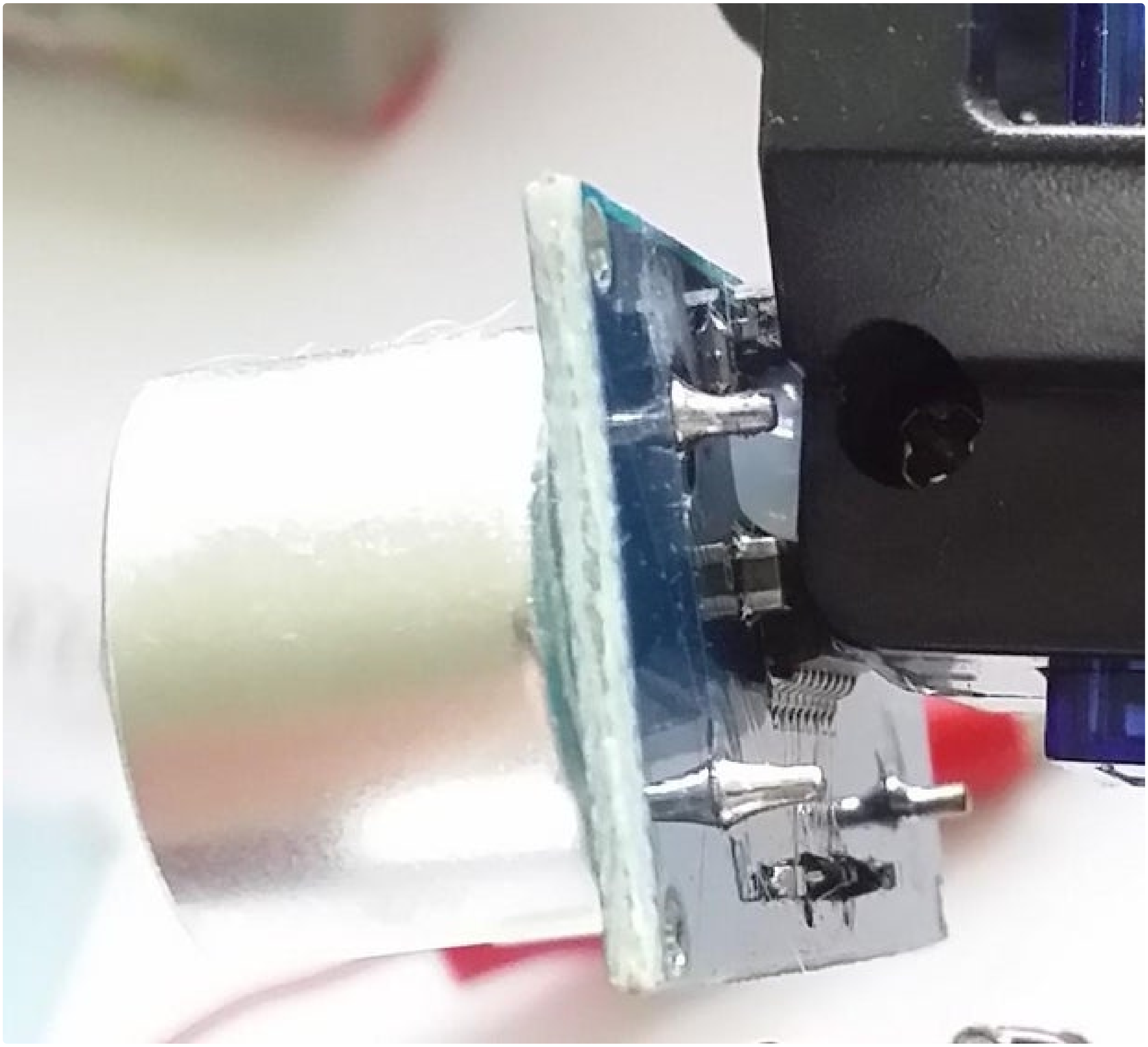




ZK - 4 asse
hurry up to te
fall from the sky, chassis materials fo
use, easy to cause the motor burned. W





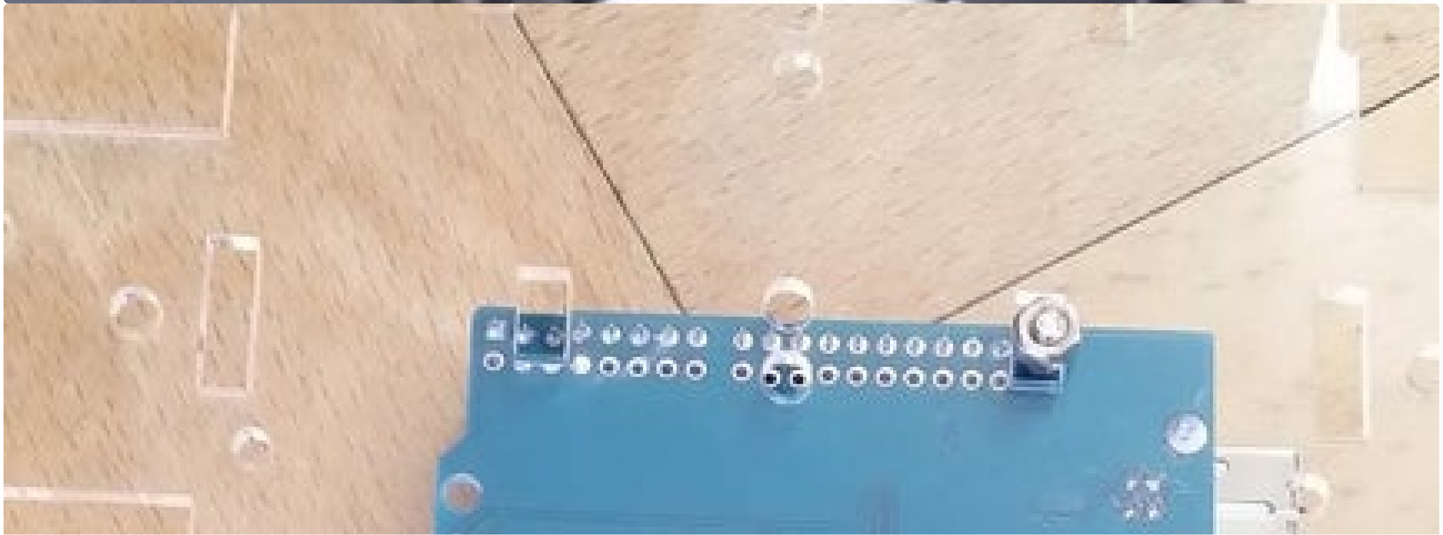


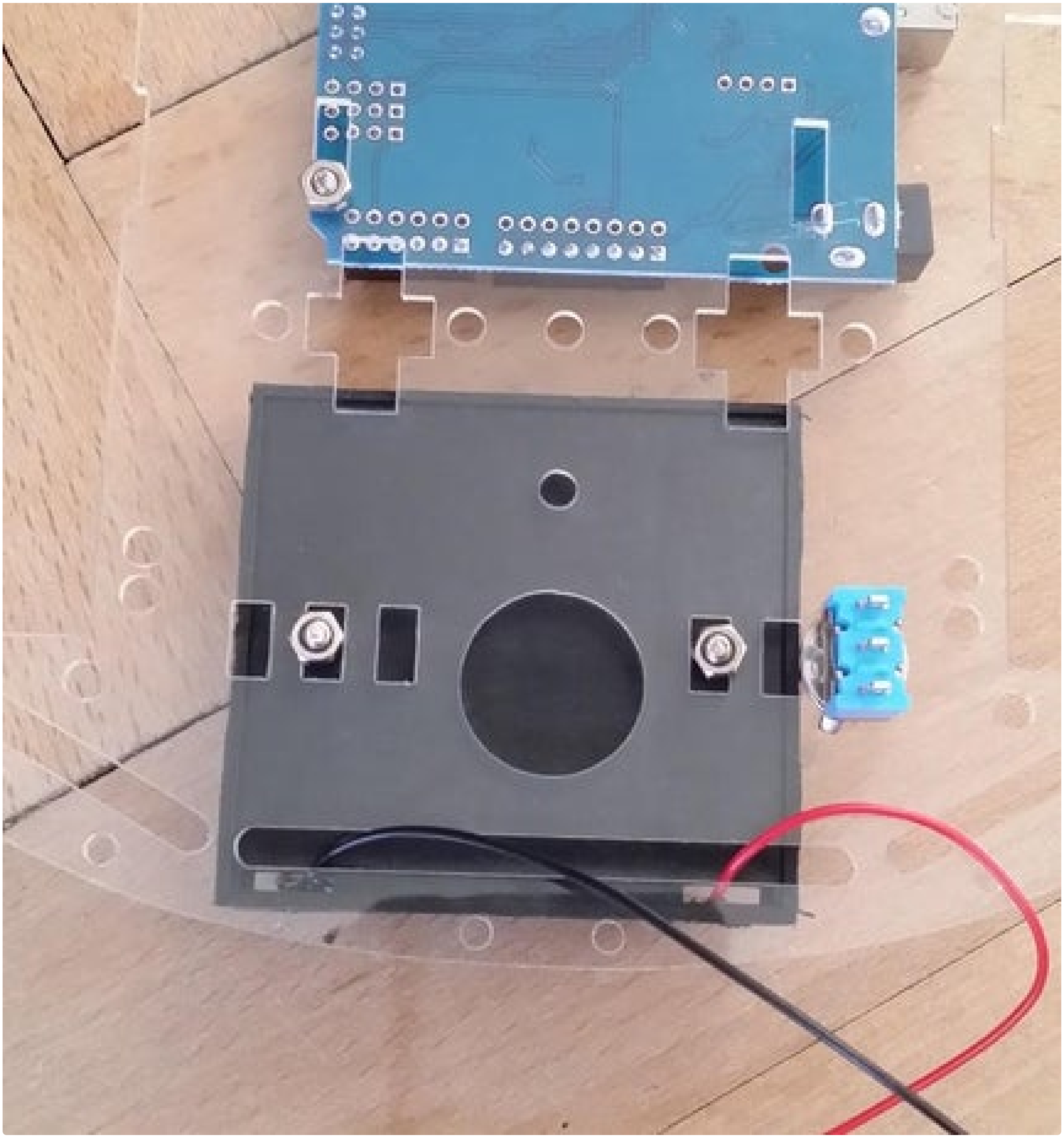
Step 4: Battery Holder and Power Switch

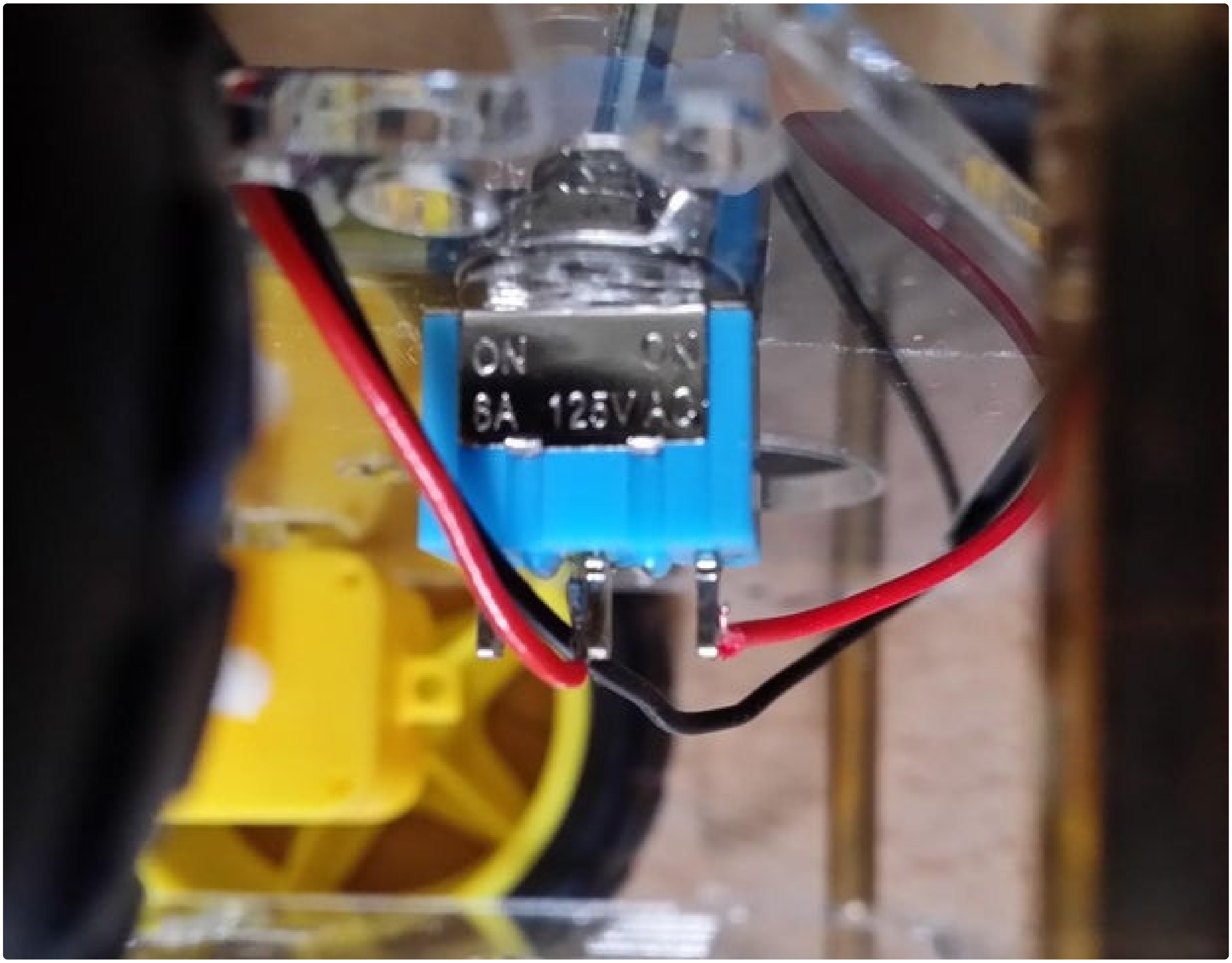
If you use the provided M3 screws to attach the battery holder, the 2 outmost batteries will no longer fit seamlessly. So use 5 mm drill to **drill an indentation for the M3 screws** and mount the holder using the rectangle holes at the back of thr top deck.

The outmost rectange hole must be widened e.g. by drill or a soldering iron :-) for a big toggle switch to fit through.

If you have the **possibility to recharge Li-ion 18650** cells, you should consider to use a battery case for two 18650 Li-ion batteries instead.







Step 5: Motor and Arduino Power Supply

Since we have 6 volt or even 4.8 volt -using NIMH batteries- as supply voltage, it is no fun to use a **L298 motor driver**, which **has a loss of 2 volt** The **TB6612 driver** breakout board costs around 3 to 4 € and **has no loss!**

Nevertheless, the **L298 driver** is supported by the motor control library too.

- To avoid mistakes while connecting theTB6612 to the UNO board, you can **remove the unused pins** of the pin headers before soldering, like in the picture.
- **To reduce additional loss**, solder the battery plus and minus cables directly to the VM(otor) and GND terminals of the **TB6612 driver** breakout board.
- If you solder the the motor wires to the (A|B)O1 and (A|B)O2 terminals you may reduce further voltage losses, but then you cannot easily separate the two decks after soldering. For this, solder the motor wires to female pin headers.
- Do not forget to **connected the STBY terminal to the VCC terminal** on theTB6612 driver breakout board, since we do not require this functionality and do not want to use an aditional Arduino output for it.
- For optional battery voltage readout, you can **solder 100 kOhm between VIN and A2 and 10 kOhm**

between A2 and ground.

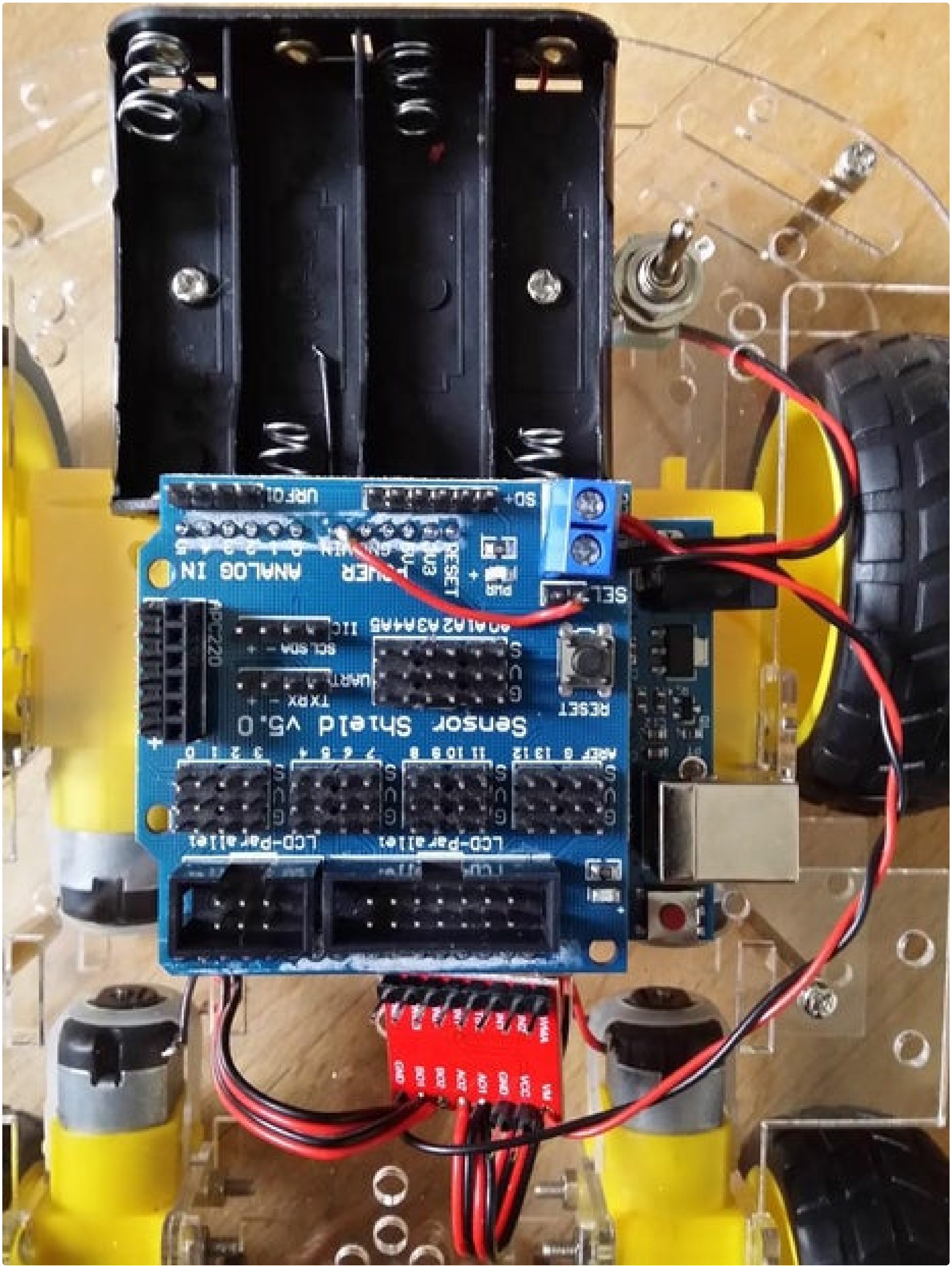
The **Arduino UNO board cannot supplied by the 6 volts of the 4 AA batteries directly**, it is only specified up to 5 volt.

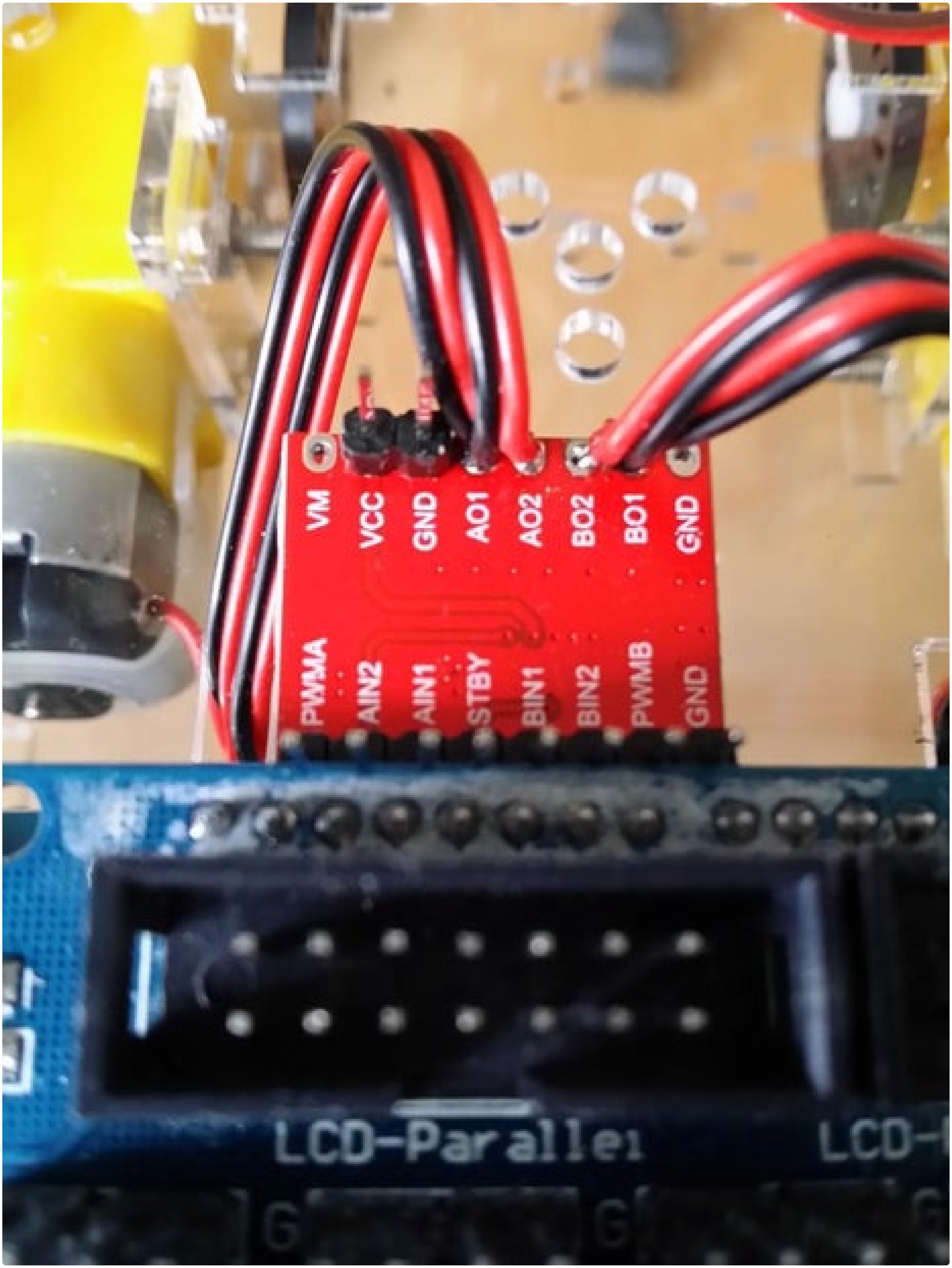
You can **use the GND and VCC terminals of the Sensor Shield** for power connection instead of the UNO power jack. The **V pins of the digital output row on the Sensor Shield are now connected with VIN** (battery) and not with the UNO VCC (5V). This improves servo performance, but may destroy any 5 volt logic circuit, if they use the V pins for power supply. **Therefore using the GND and VCC terminals is not recommended for a 2 LI-ion supply, in this case you must use the UNO power jack!** Otherwise it will likely destroy the SG90 servo, which is specified up to 6 volt.

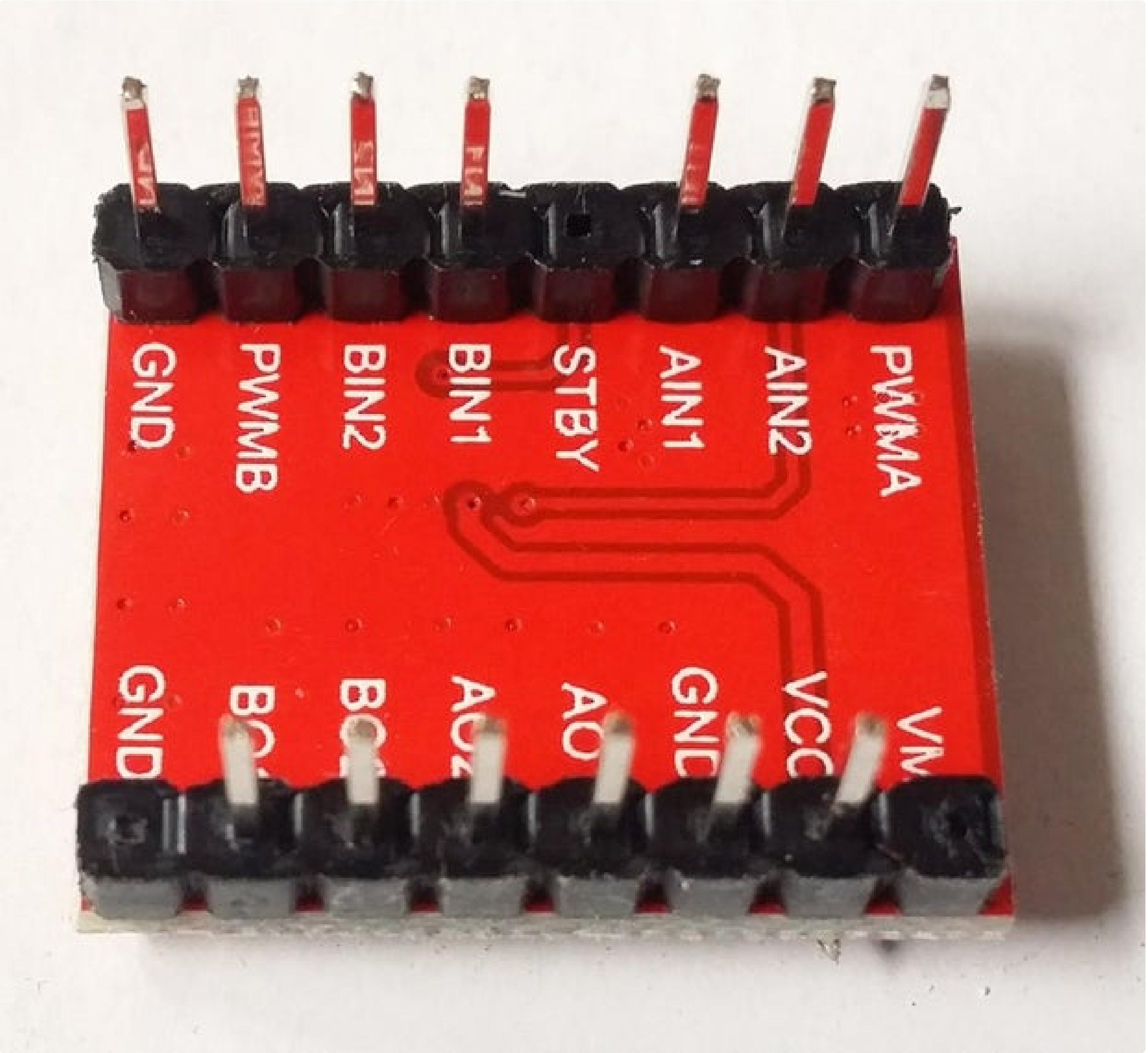
- **Remove the SEL bridge**, which connects the VCC with the internal 5 volt rail of the Arduino.
- Connect the **external VCC terminal with the Arduino UNO VIN** pin.

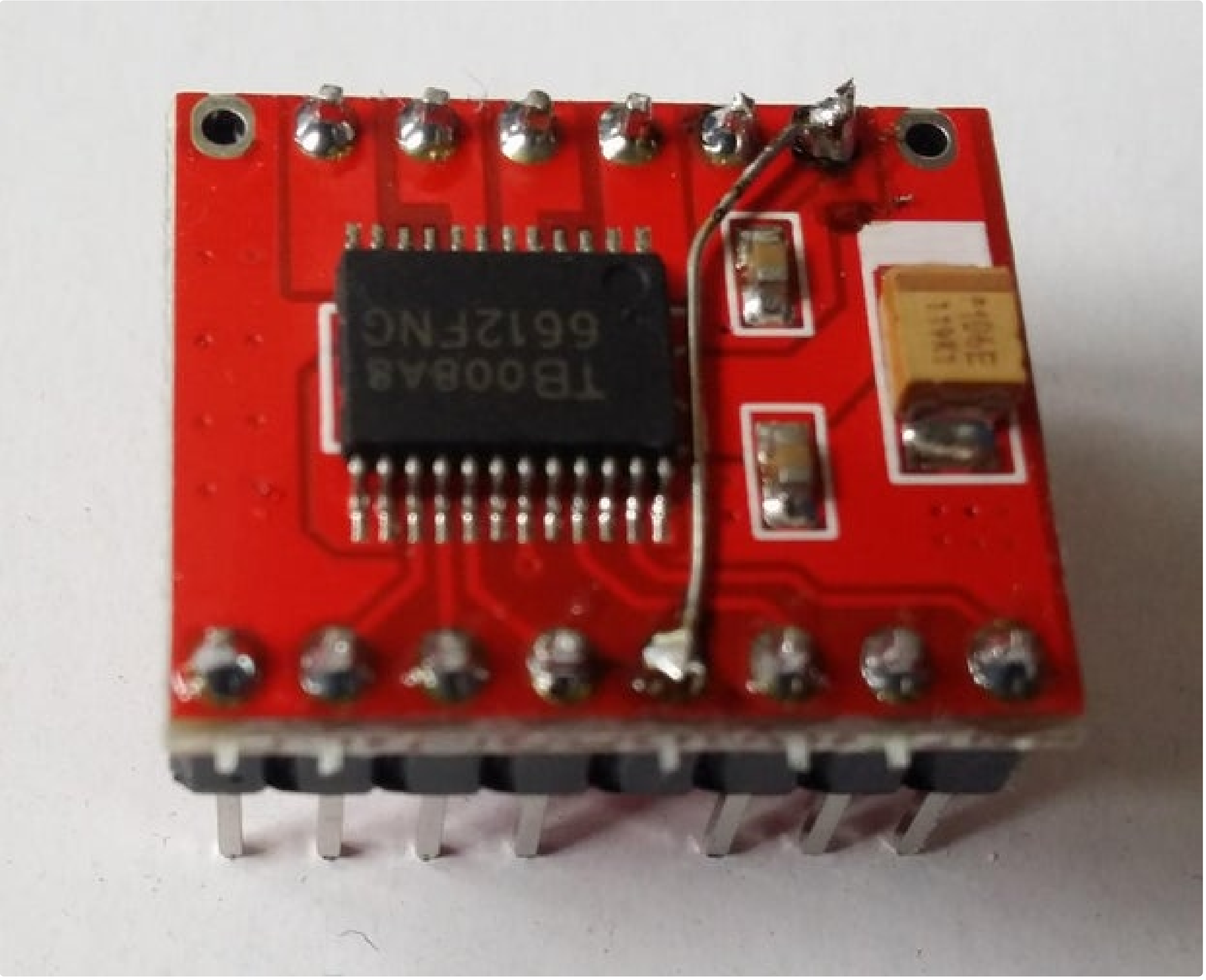
The Arduino is supplied with VIN - 0.9 volt, and runs down to aproximately 3.5 volt.

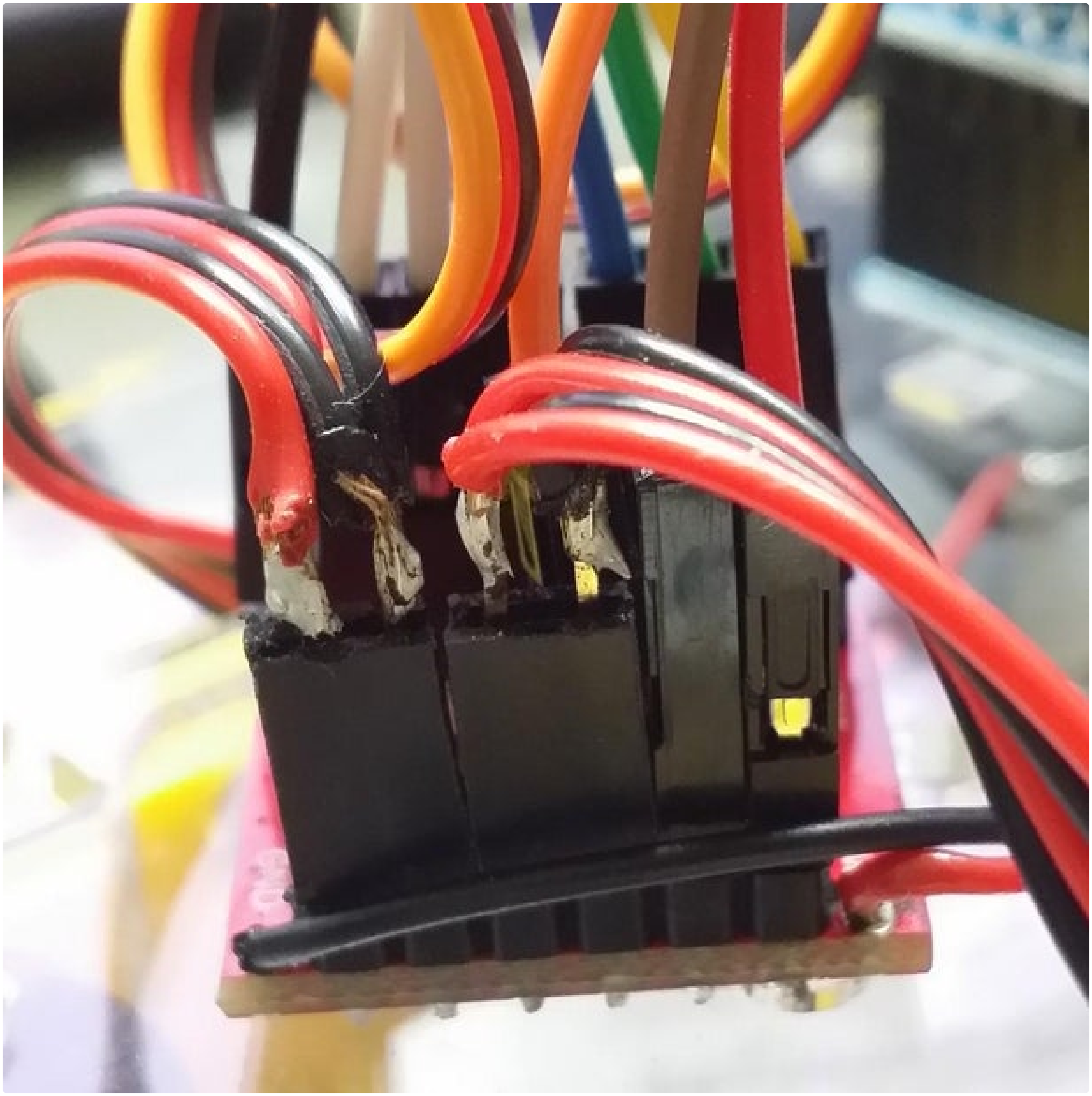
If you are using **4 NIMH rechargeable AA batteries**, you should **leave the SEL bridge**, which connects the VIN with the Arduino 5 volt. **This avoids the 0.9 volt loss** of the 5 volt regulator and some resets if the batteries are weak and the motors start.


Sensor Shield v5.0
RESET
LCD-Parallel
LCD-Parallel
ANALOG IN
POWER
URF01
SD+
SEL
APC220
IIC
UART
PWR


VM
VCC
GND
AO1
AO2
BO2
BO1
GND
PWMA
AIN2
AIN1
STBY
BIN1
BIN2
PWMB
GND
LCD-Parallel
LCD-


GND
PWMB
BIN2
BIN1
STBY
AIN1
AIN2
PWMA
GND
BO
BO
AO
AO
GND
VCC
VM



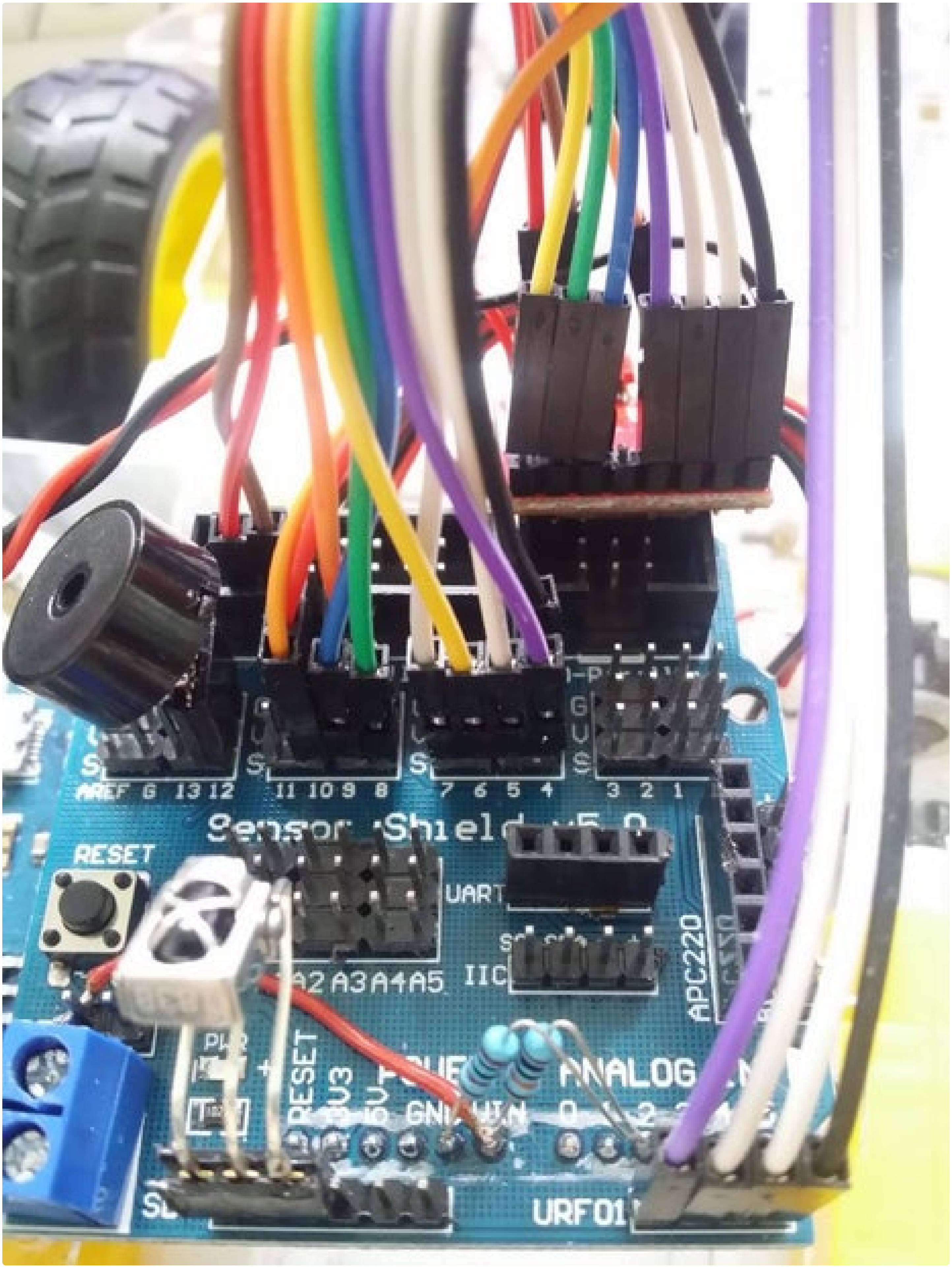


Step 6: Control Connections

The actual connection between the Sensor Shield and the servo, the ultrasonic sensor and the motor driver is of course software dependent. Here are the definitions used by the PWMMotorControl Arduino library. The line marked with *AVR (UNO)* (line 37) shows the relevant connections for us.

To connect the motor driver you can use the 15 cm cables inteded for the IR obstacle detection curcit or use extra12 cm female-female dupont wires.

Do not forget to connect the Arduino VCC and GND with the VCC and GND terminals of the TB6612 board.


Sensor Shield v5.0
RESET
AREF G 13 12
11 10 9 8
7 6 5 4
3 2 1
G
V
S
UART
IIC
APC220
A2 A3 A4 A5
RESET
3V3
5V
GND
VIN
ANALOG
0 1 2
URF01

https://www.instructables.com/ORIG/FJ6/H1O4/L1HV9LH3/FJ6H1O4L1HV9LH3.h

Download

Step 7: Motor Test

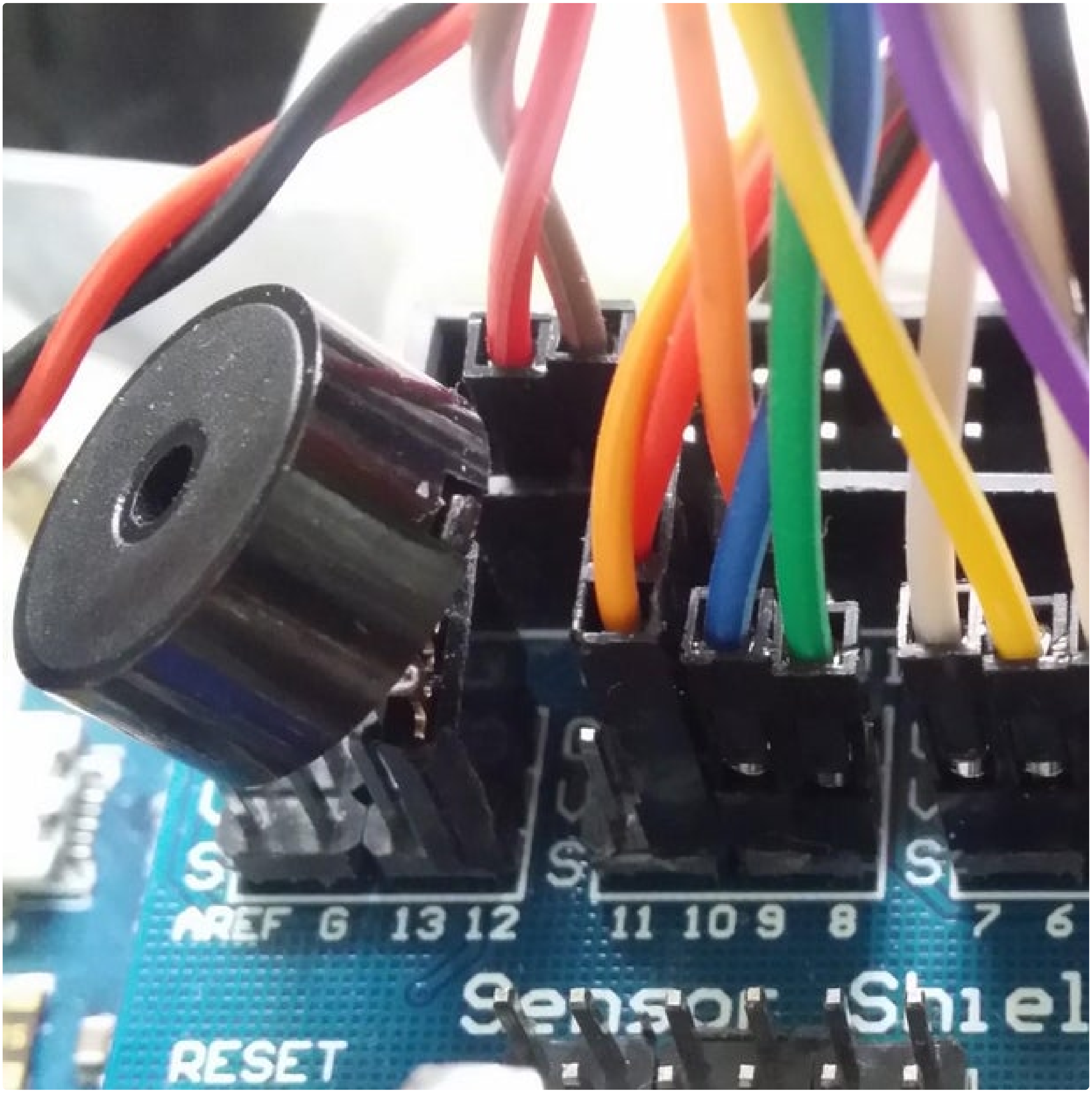
- To test your work, you can attach a piezo at pin 12 for acoustic boot and distance feedback.
- Upload the Start example of the Arduino PWMMotorControl library to the UNO board.
- **Observe the serial output** (at 115200 baud) and compare it with the actual movements of the car wheels.

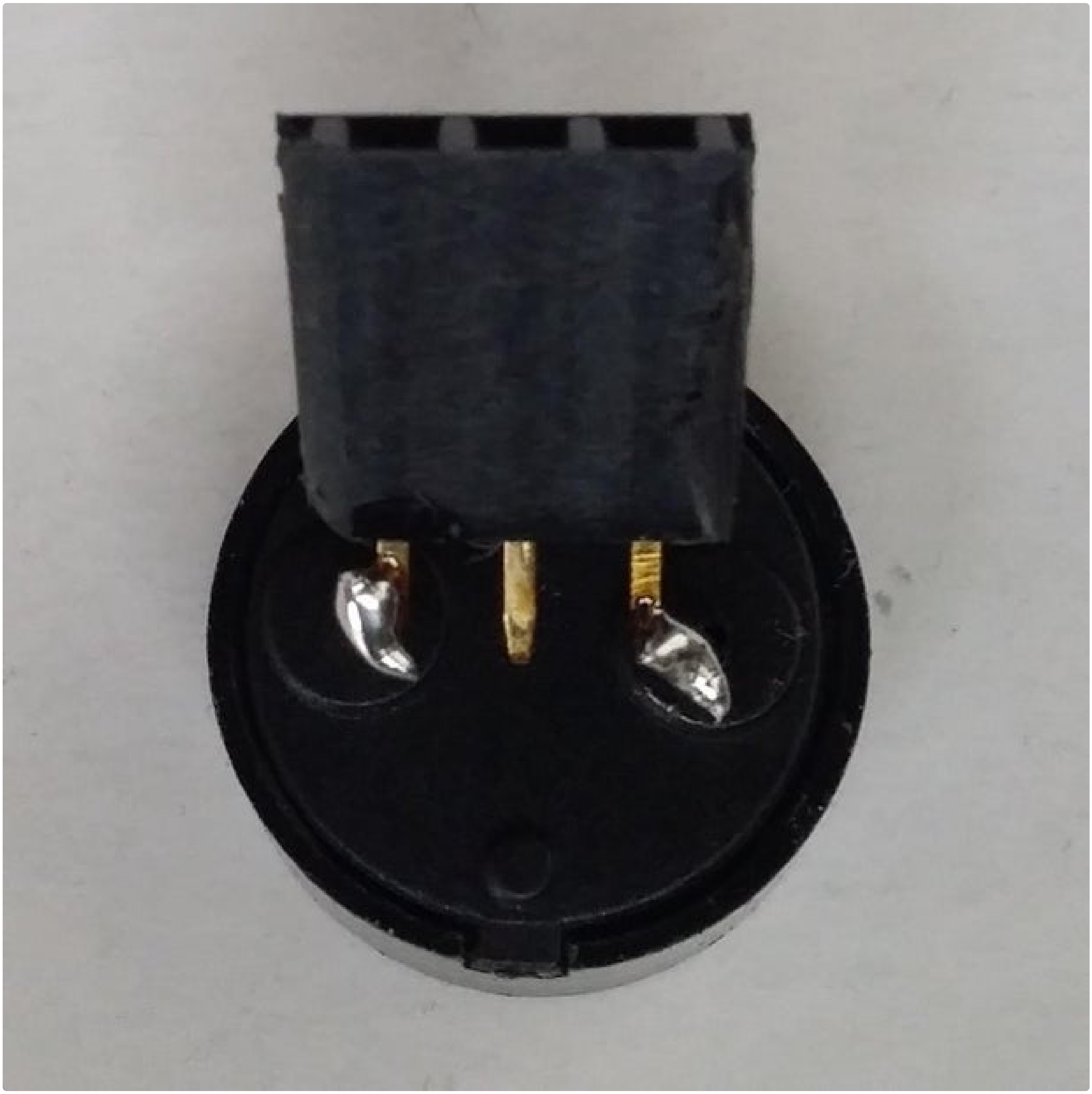
If you see errors like this

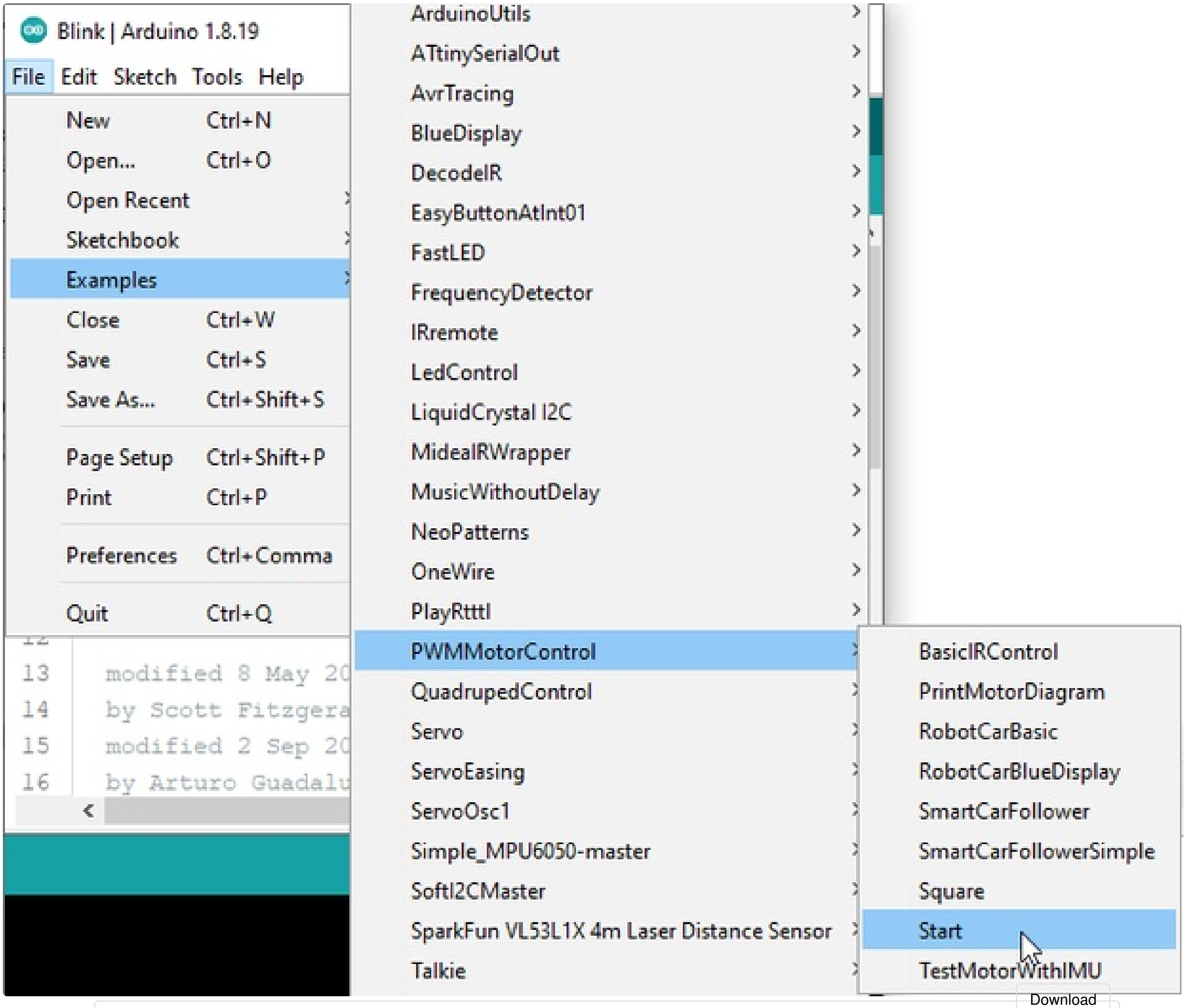
```
avrdude: verifying ...
 avrdude: verification error, first mismatch at byte 0x006c
 0xc4 != 0xd4
 avrdude: verification error; content mismatch
```

during upload try to use another USB connection cable or to temporarily disconnect the Sensor Shield from the UNO board or switch the battery power on while uploading.

If the motors rotate in the wrong direction, you have to swap the motor connections (A,B)O1 and (A,B)O2, or the (A,B)IN1 and (A,B)IN2 connectors on the TB6612 board.


AREF G 13 12
11 10 9 8
7 6
S
S
S
Sensor Shiel
RESET





```
modified 8 May 20
by Scott Fitzgera
modified 2 Sep 20
by Arturo Guadalu
```

Download

https://www.instructables.com/ORIG/F1J/2H90/L1M5K7GG/F1J2H90L1M5K7GG.ino

Step 8: Distance Sensor and Servo Test

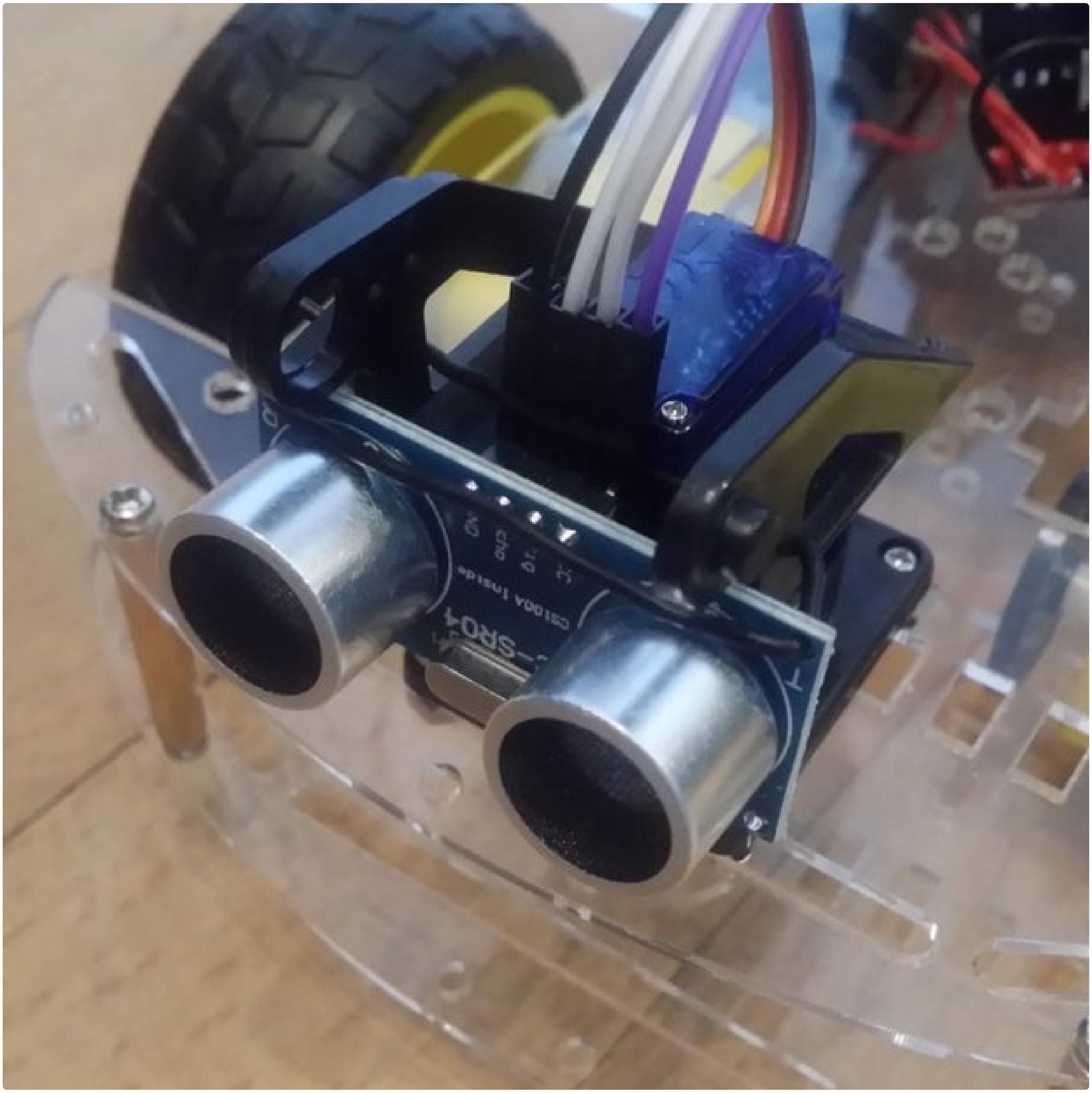
To test your distance sensor, upload the SmartCarFollowerSimple example to the UNO board.

After booting, the servo should move to 135 and 45 degree and then back to 90 degree.

For distances above 40 cm you should hear a tone of **1500 Hz**, for distances below 30 cm **666 Hz**, and for distances between **1000 Hz**.

The car moves forward and backwards in order to **keep a distance between 30 and 40 cm to the obstacle**.

Take care, that **the line with the right configuration** e.g. **#define TBB6612_BASIC_4WD_4AA_CONFIGURATION** is activated before compile!


-SR04
CS100A Inside

Download

https://www.instructables.com/ORIG/FEM/HH8T/L21VFN3O/FEMHH8TL21VFN3O.ino

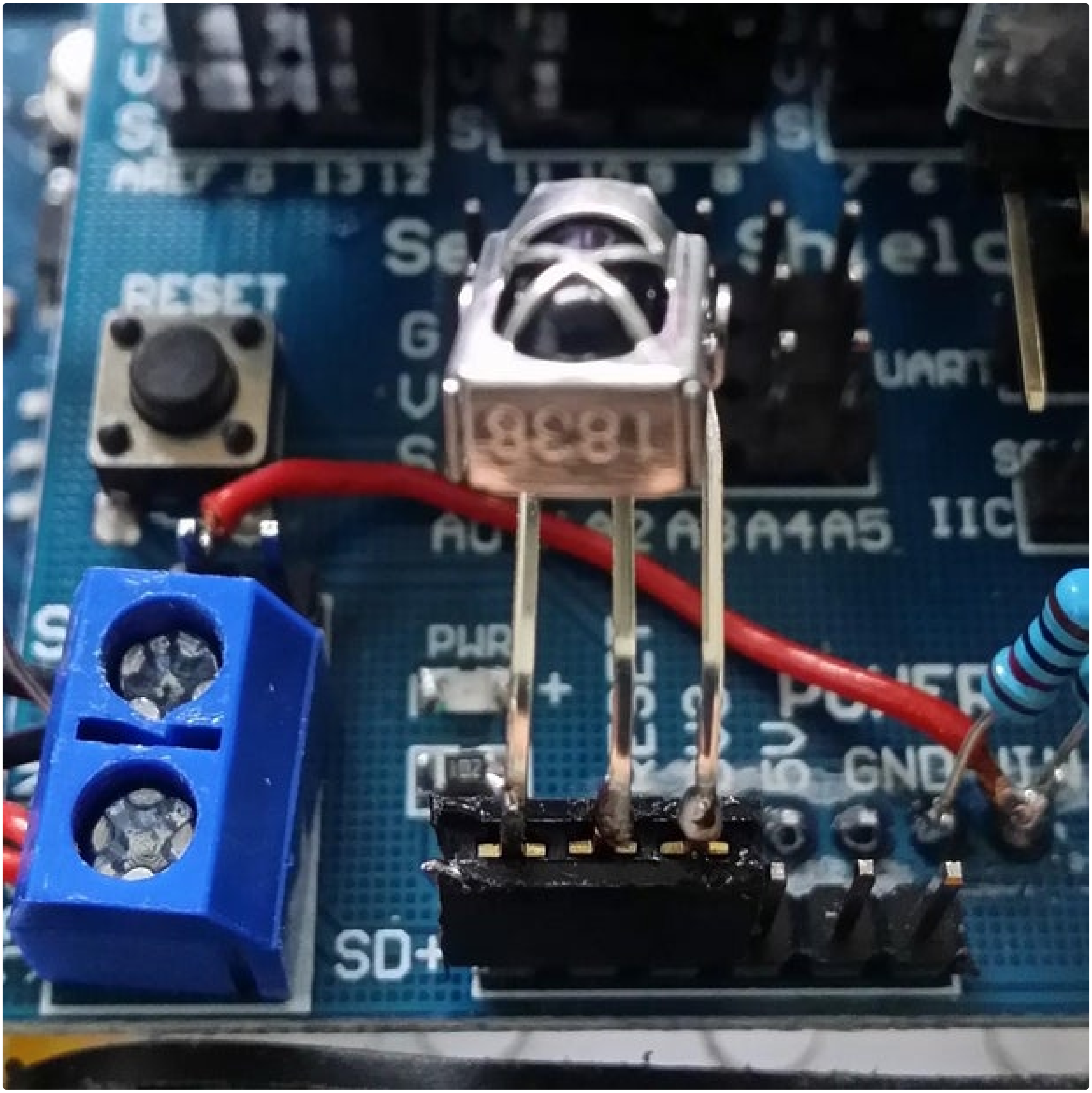
Step 9: Basic IR Control

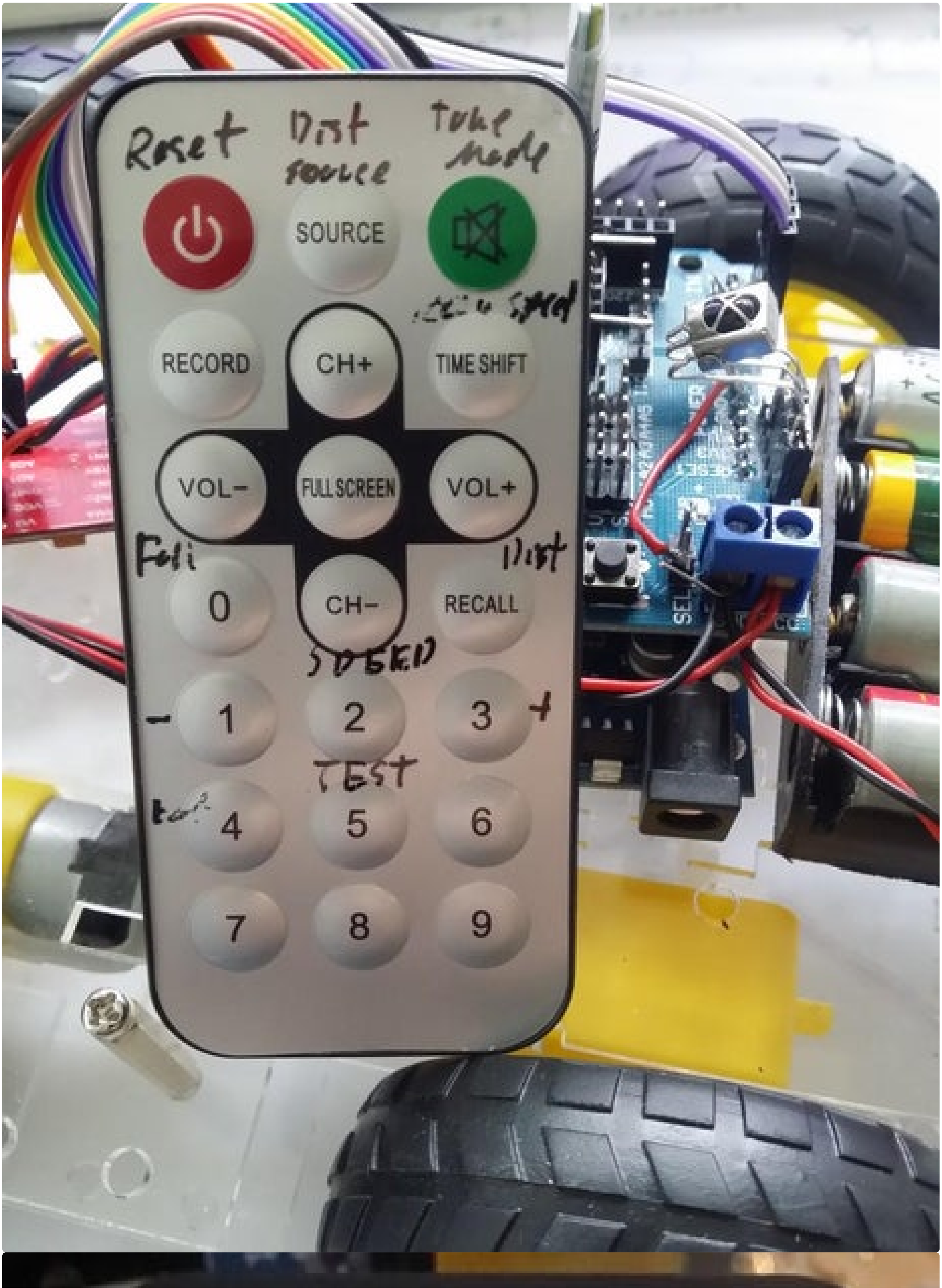
Attach the TL1838 IR receiver module to the **SD+ connector** of the Sensor Shield. **Do not use the row, where the buzzer and the motor driver is attached**, since the middle V pin there is connected with VIN, which may have 6 volt and therefore will destroy the TL1838.

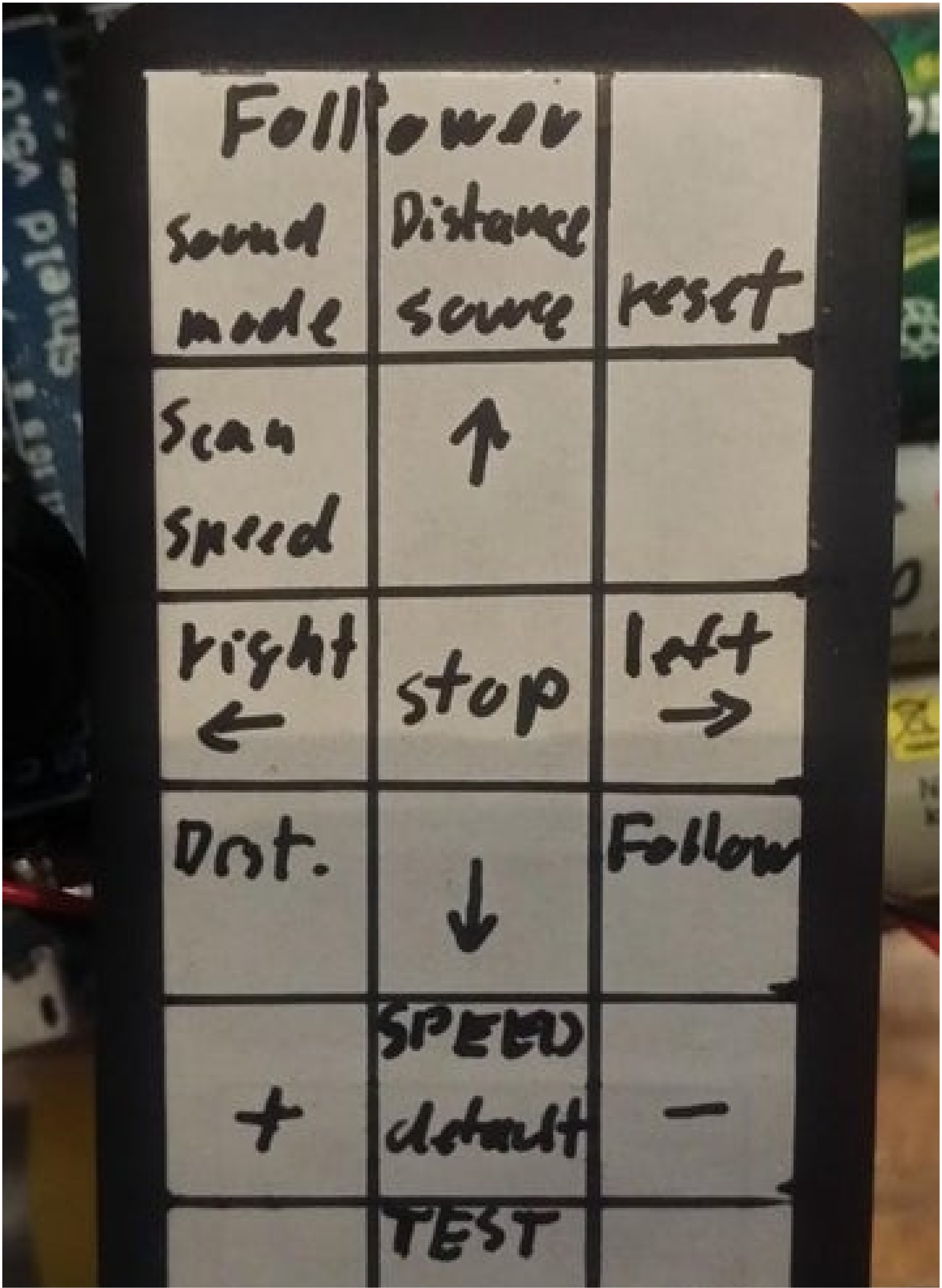
Basic IR-remote car control like move and turn is implemented in the BasicIRControl example.

Mapping between keys of **any arbitrary IR remote** sending NEC protocol (all the cheap china ones) and car commands can easily be achieved in IRCommandMapping.h.

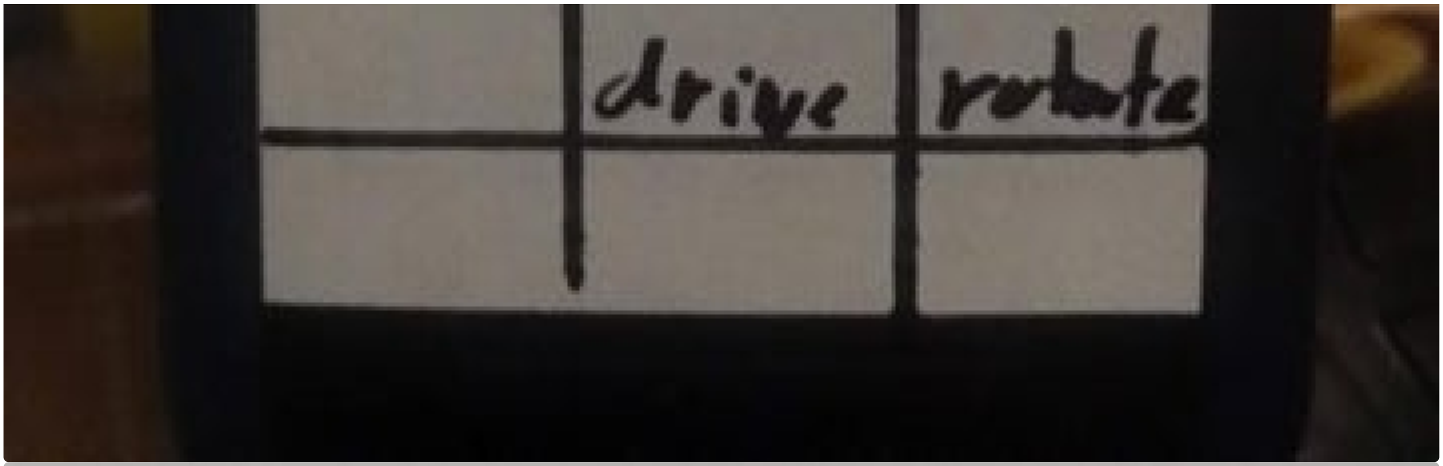
To support your mapping, the received IR code is printed at the serial output if `INFO` is defined (default).

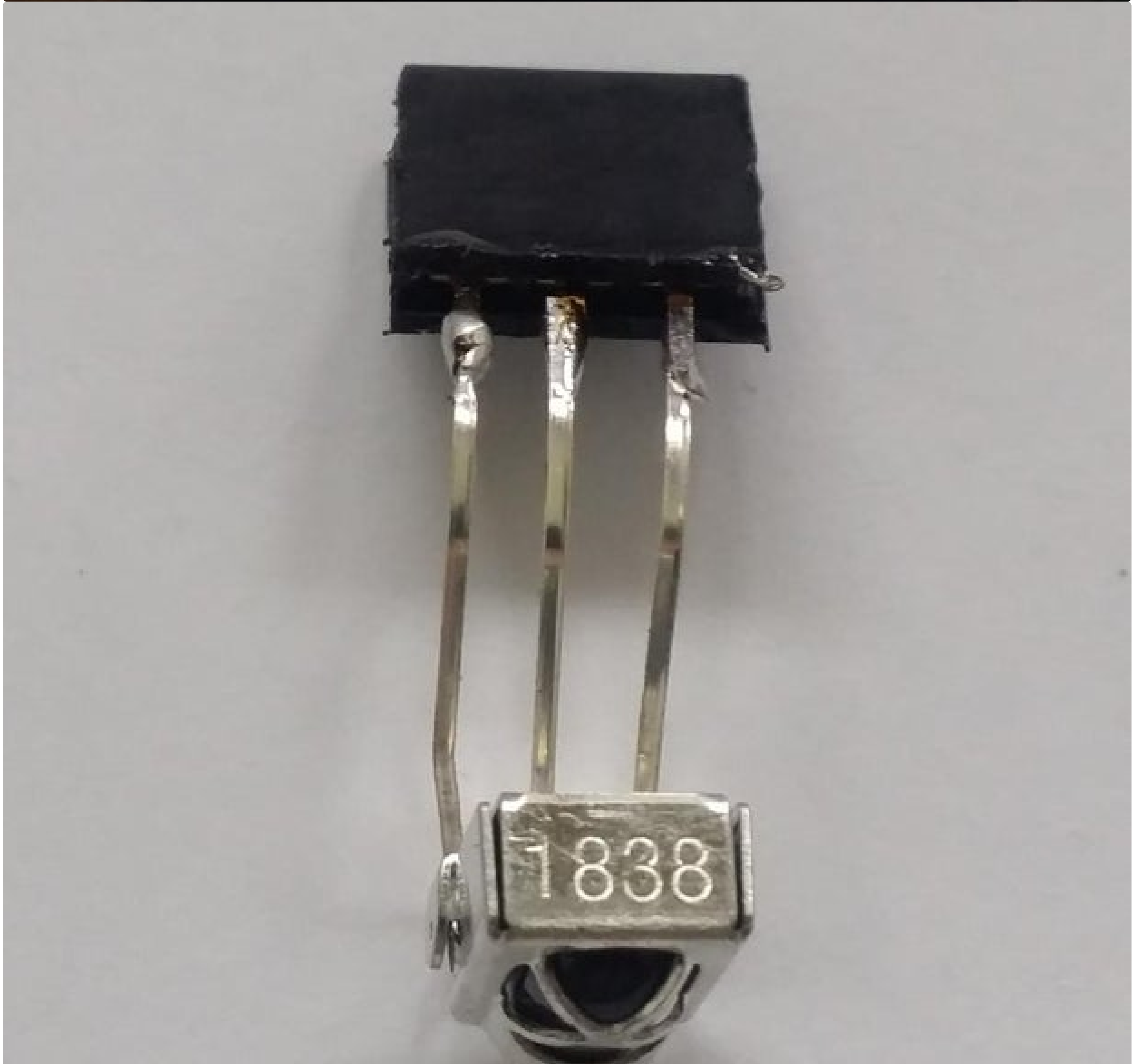

RESET
1838
UART
IIC
A4A5
PWR
POWER
5V
GND
SD+


Reset
Dist source
Tune mode
SOURCE
Speed
RECORD
CH+
TIME SHIFT
VOL−
FULLSCREEN
VOL+
Foli
Dist
0
CH−
RECALL
SPEED
−
1
2
3
+
TEST
4
5
6
7
8
9



Foll Sound mode	ower Distance source	reset
Scan Speed	↑	
right ←	stop	left →
Dist.	↓	Follow
+	SPEED default	−
	TEST	


drive
rotate


1838

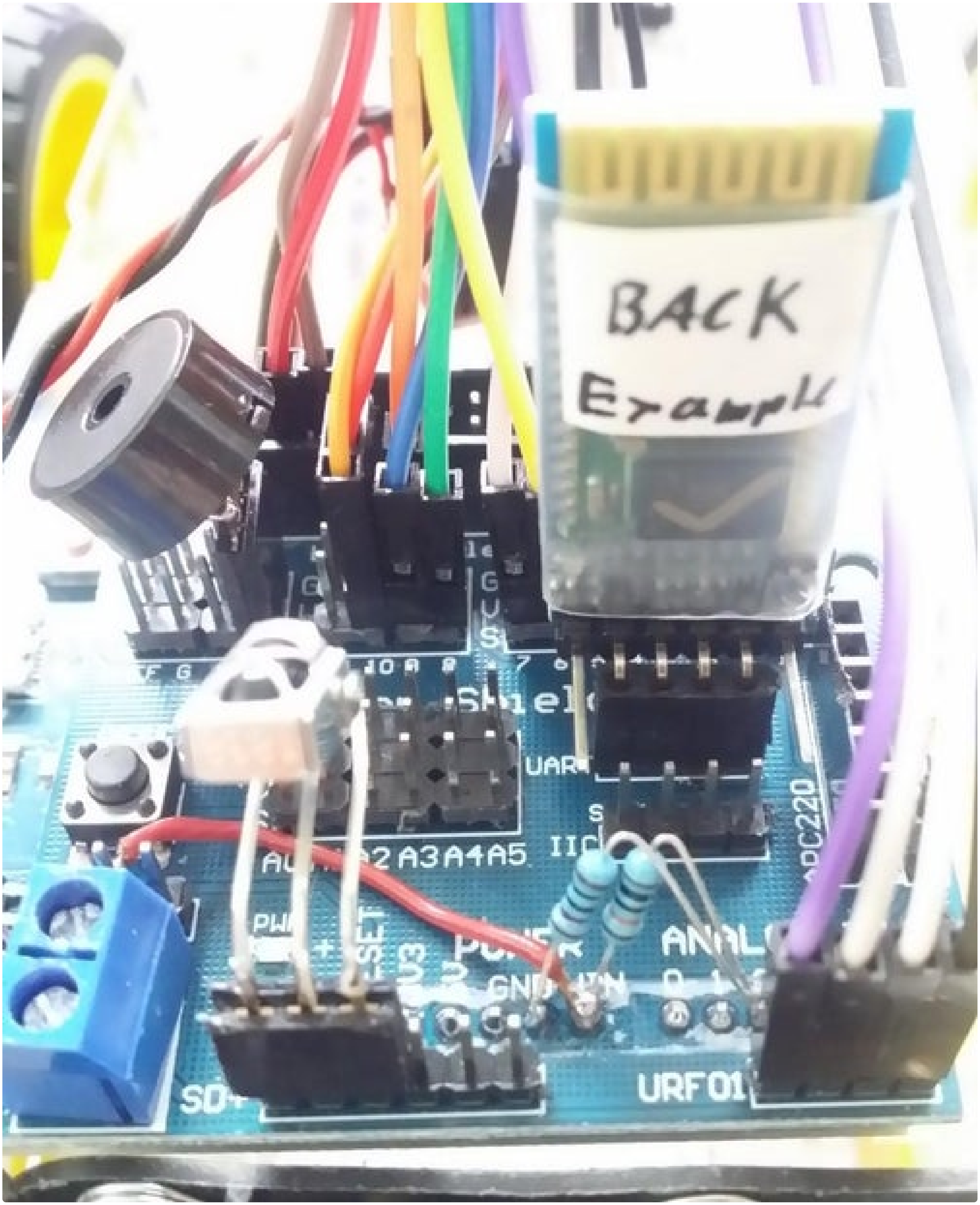
https://www.instructables.com/ORIG/FSC/XR4X/L1M5K7II/FSCXR4XL1M5K7II.ino Download

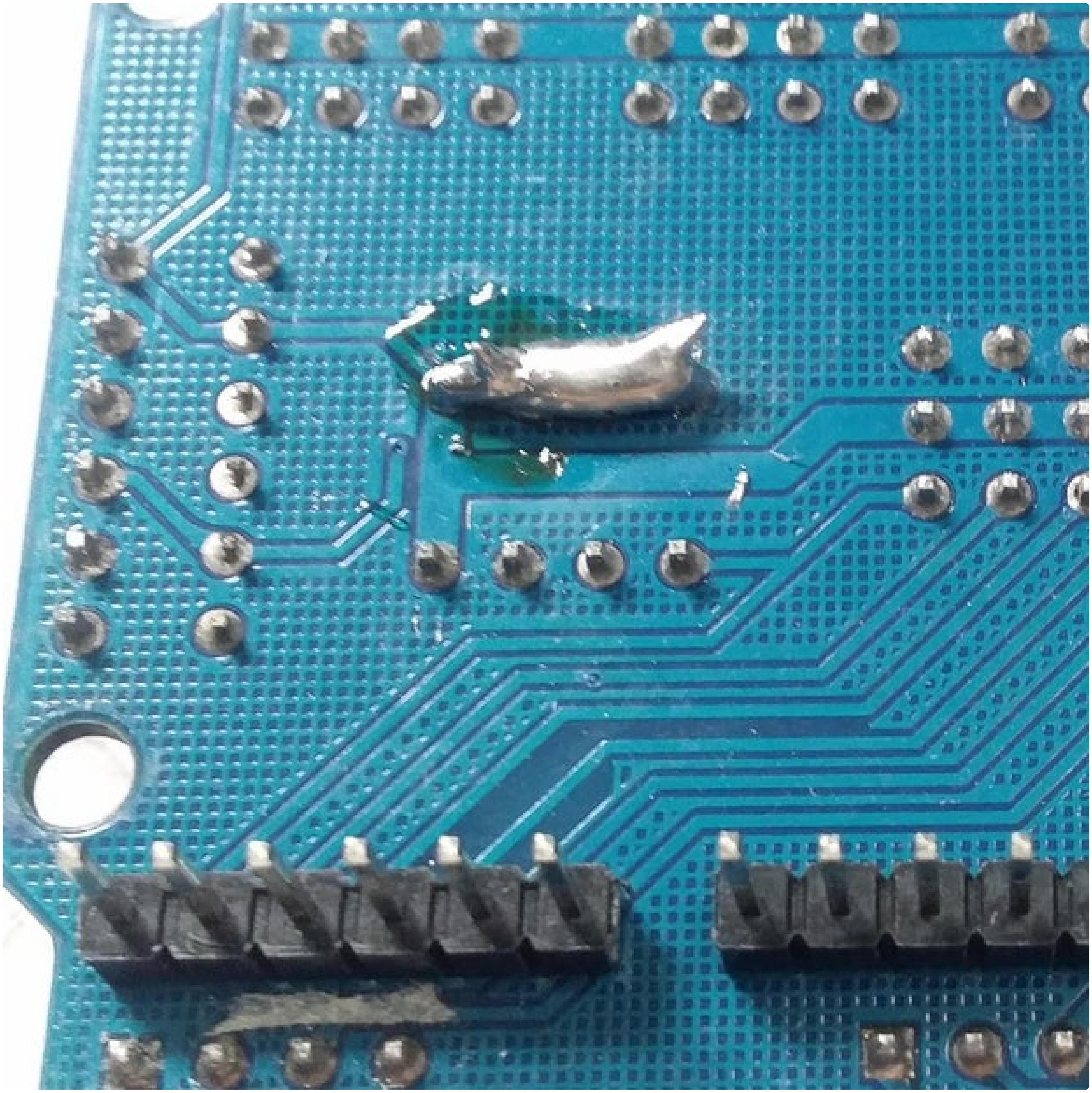
https://www.instructables.com/ORIG/F7J/OL26/L1M5K7MX/F7JOL26L1M5K7MX.h Download

Step 10: Modify SensorShield for Bluetooth Module HC05

For Bluetooth control, I use a HC05 bluetooth modul and connected it to the UART pins of the Sensor Shield.

To avoid another set of cables, I changed the original male UART pin header row to a female row. Desoldering (and soldering) can be done by heating all 4 pins simultaneously, with the help of a lot of solder.


BACK
Example
UART
IIC
A0 A1 A2 A3 A4 A5
APC220
PWR
POWER
GND VIN
ANALOG
URF01
SD



Step 11: Smartphone Control

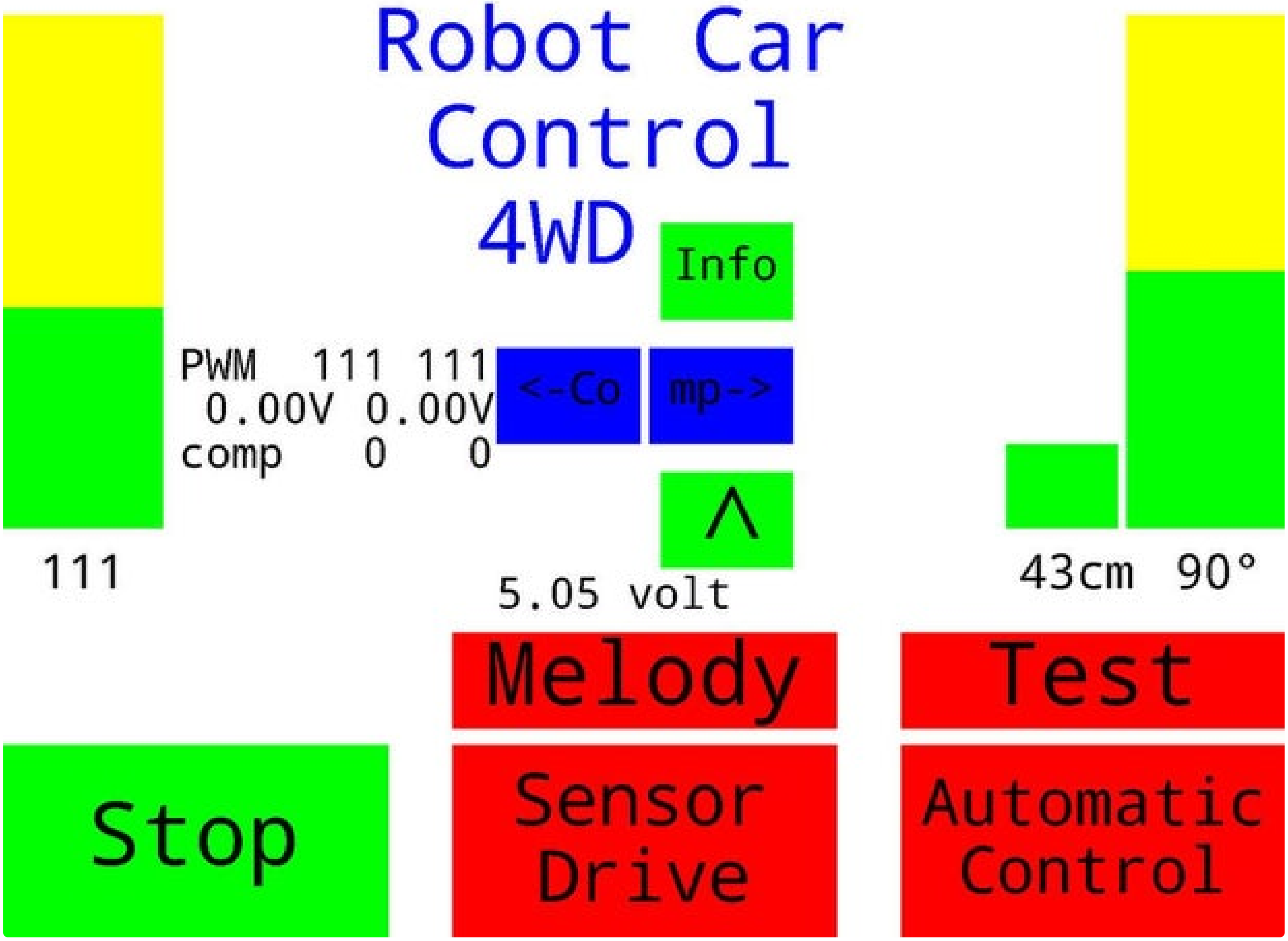
If you have added a bluetooth module, you can **control your car by your smartphone**using the RobotCarBlueDisplay example of the BlueDisplay Arduino library. On your **Android smartphone** you need to install the Android BlueDisplay app.

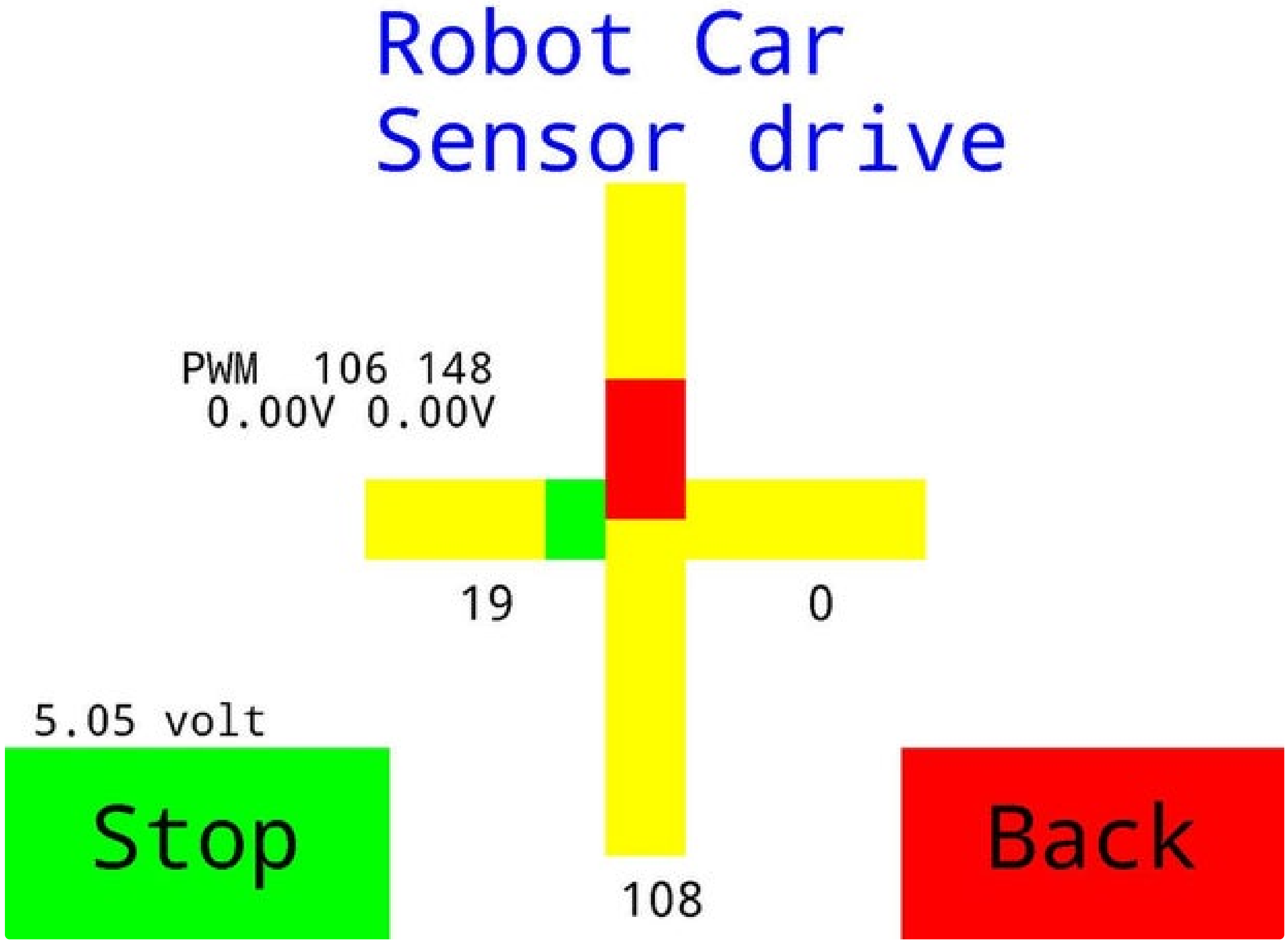
The Bluetooth module can be modified by means of the BTModuleProgrammer example from the BlueDisplay library to have a unique name and **to support 115200 baud**.

Do not forget to **remove the Bluetooth module during programming of the UNO**, otherwise you will get errors like:

```
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0xa5
```

Take care, that **the line with the right configuration** e.g. **#define TBB6612_FULL_4WD_4AA_CONFIGURATION** is activated before compile!

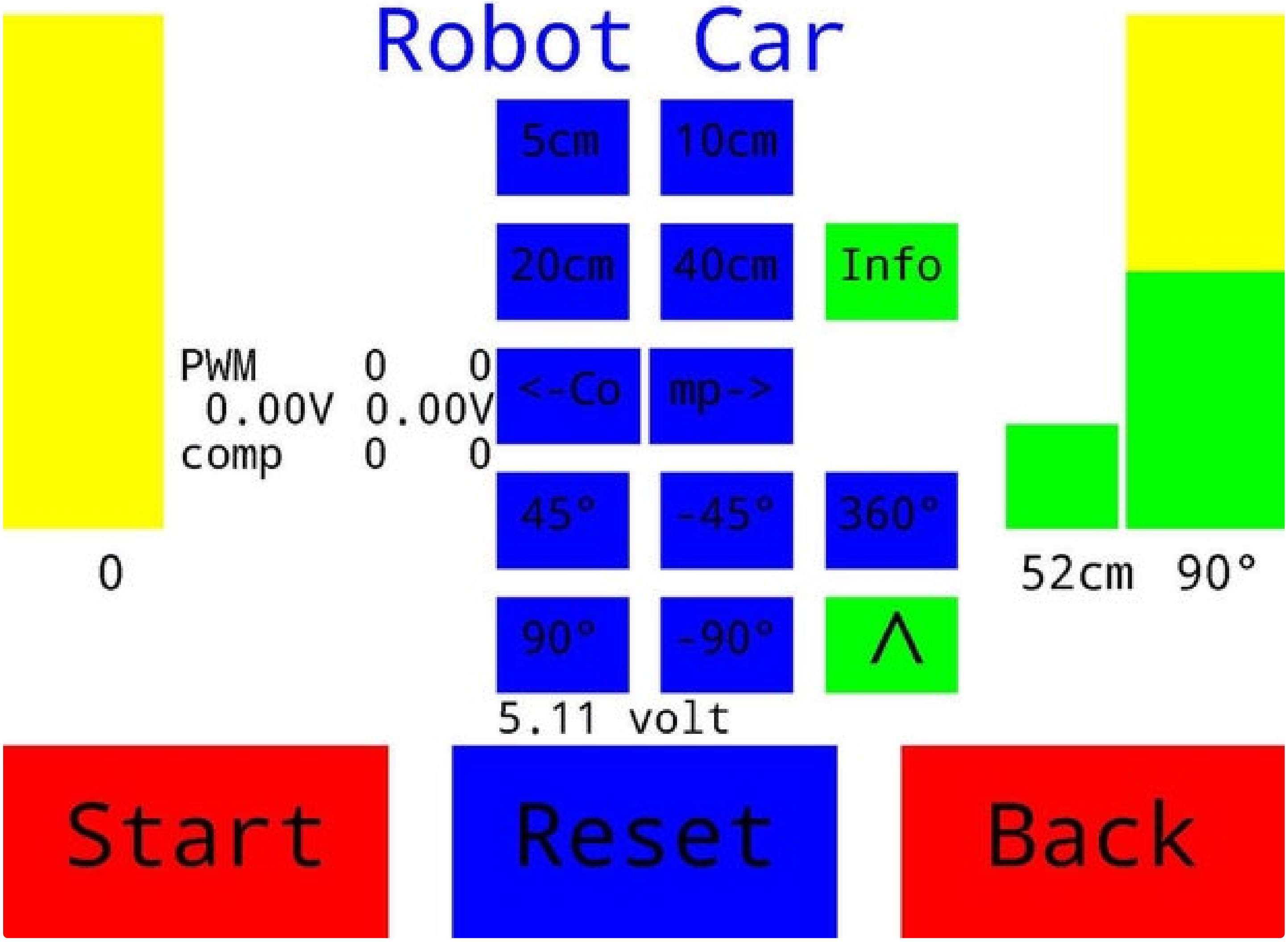




Robot Car
Sensor drive

```
PWM  106 148
 0.00V 0.00V
```

5.05 volt

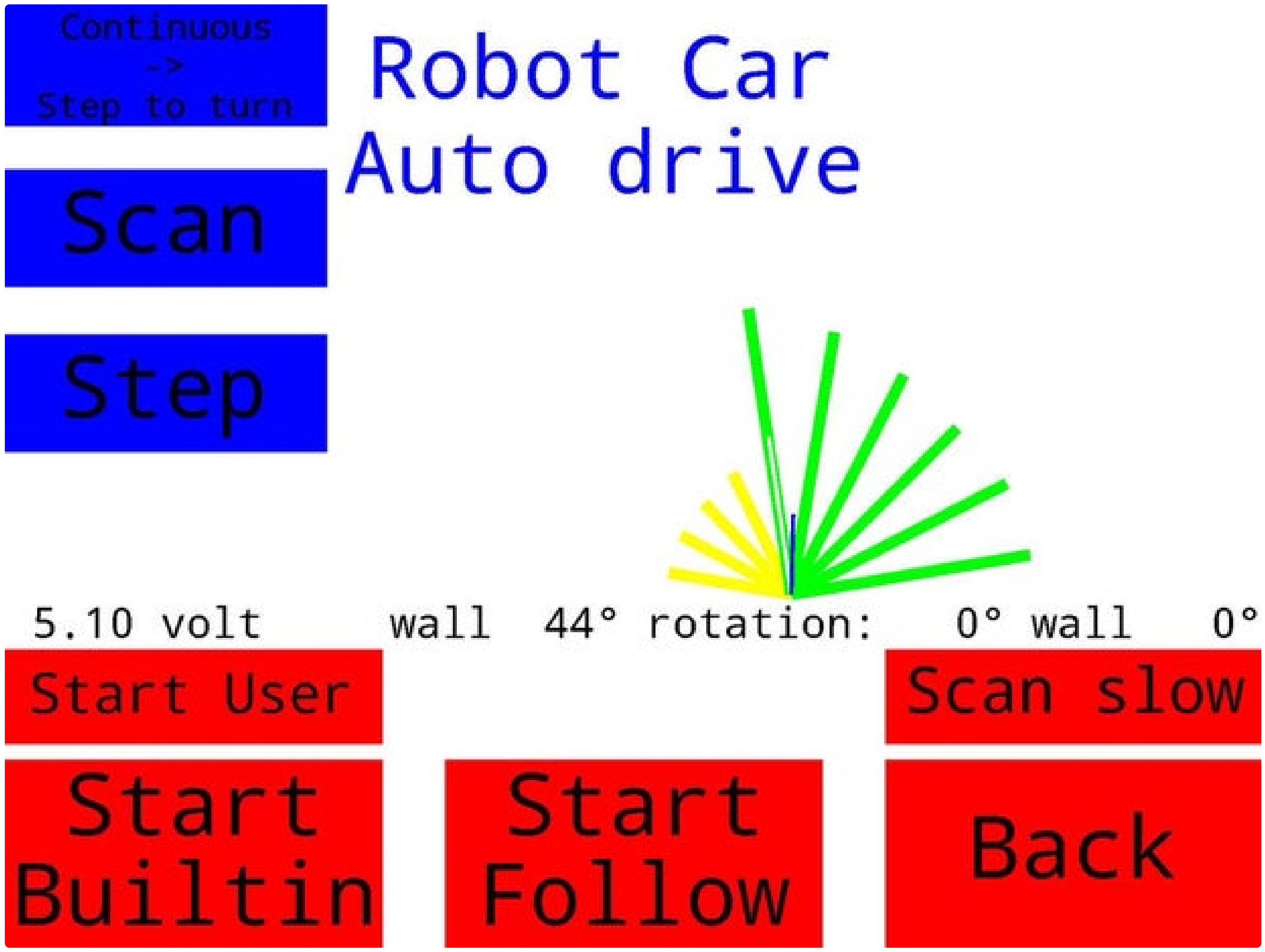


```
PWM      0    0
 0.00V 0.00V
comp     0    0
```

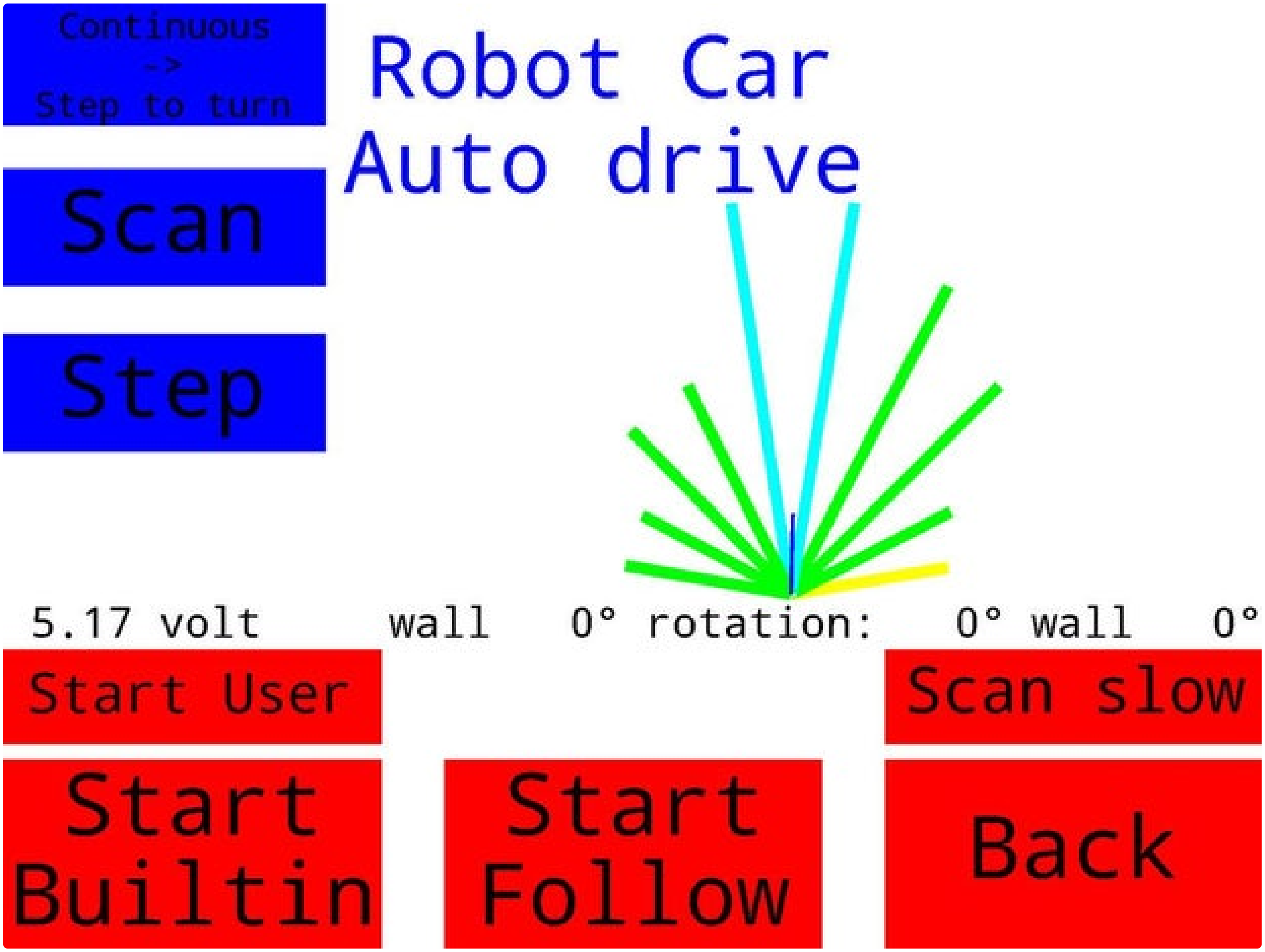
0

52cm 90°

5.11 volt



```
5.10 volt     wall  44° rotation:   0° wall   0°
```



```
5.17 volt     wall   0° rotation:   0° wall   0°
```

Step 12: Deterministic Turns and Distances With MPU6050 IMU

The Arduino library supports usage of a MPU6050 IMU module to go fixed distances and do determinist turns. But it is still not fully tested for this type of robot car.

Step 13: Remarks

I hope you have fun with the car, I especially enjoy the sensor drive function with bluetooth.

This cars and the library are for my Arduino AG class at the Montessory Gymnasium.

This Instructable and the library are still work in progress (5/2022).

Maybe I can add a few videos in the next month.

If you have improvement requests, do not hesitate to use the github discussions function of the library.

It would be a pleasure to hear from you, if you use this instructions or the library for your own projects.

Armin