Development of Arduino-based robot curriculum for Electronics Engineering Technology

2025 Summer Enhancement Grant

Erik Mayer

Electronics Engineering Technology and Master of Engineering Technology programs

Motivation

- Retain Freshman students
  - Use robots in EET 100 on 11/18 and 11/20
- Revise Curriculum
  - Use robots in EET 344 by end of Fall 2025 semester
- Recruit students
  - This Four-State presentation
  - Future Teacher workshops?

Introduction

The development of Arduino-based robot curriculum for the Electronics Engineering Technology (EET) program which was done in three phases.

1. The first phase was the training in the fundamentals of the Arduino integrated development environment (IDE) and preparing for Arduino Certification.
2. The second phase was the investigation into Arduino Alvik robots.
3. The third phase was the retrofitting of previously purchased Texas Instruments (TI) robots with discontinued microcontrollers to use Arduino hardware.

Bachelor of Science in Engineering Technology Degree with a Major in Electronics Engineering Technology (EET)

- For the most up-to-date info, see catalog page
- Four emphasis areas
  - Automation
  - Computer and Embedded Systems (proposed name change: AI + Embedded Systems)
    - EET-449 Programmable Logic Devices (3 hours)
    - EET-549 Advanced Microcontrollers (3 hours)
    - EET-647 Digital Signal Processing (3 hours)
    - An additional upper division, above 300 level, Electronics Engineering Technology emphasis area.
  - Power Systems
  - Custom

Bachelor of Science in Engineering Technology Degree with a Major in Electronics Engineering Technology (EET)

- Accelerated Program for Master of Engineering Technology
  - Courses can be taken that count towards both EET and Master of Engineering Technology (MET)
    - EET-742 Programmable Logic Devices (3 hours)
    - EET-743 Advanced Engineering Electromagnetics (3 hours)
    - EET-745 Advanced Microprocessor Systems and Applications (3 hours)
- Minors
    - Automation Engineering Minor
    - Electronic Technology Minor
- Certificates
    - Automation Certificate

Disclaimer

- Note: that this work is experimental and in its preliminary, investigative, and testing phase.
  - Experienced past issues with “latch-up” in microcontrollers (Latch-up – Wikipedia)
    - Microcontroller became hot when voltage applied to input before microcontroller was powered
  - USB powers motors even if batteries not connected
  - Exposed battery terminals

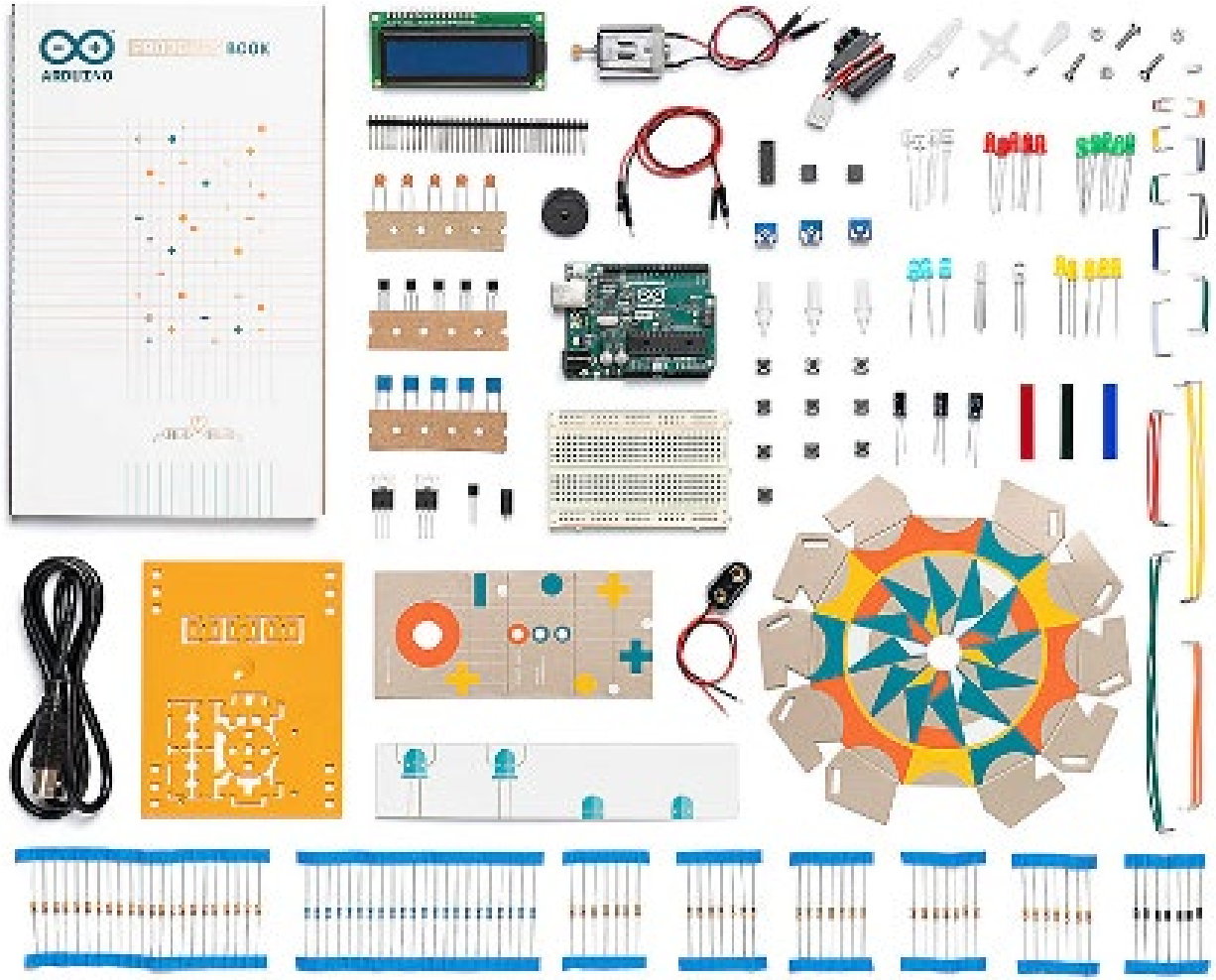
Phase 1a: Arduino Starter Kit

https://store.arduino.cc/products/arduino-starter-kit-multi-language

$88 (price on 11/9/2025)

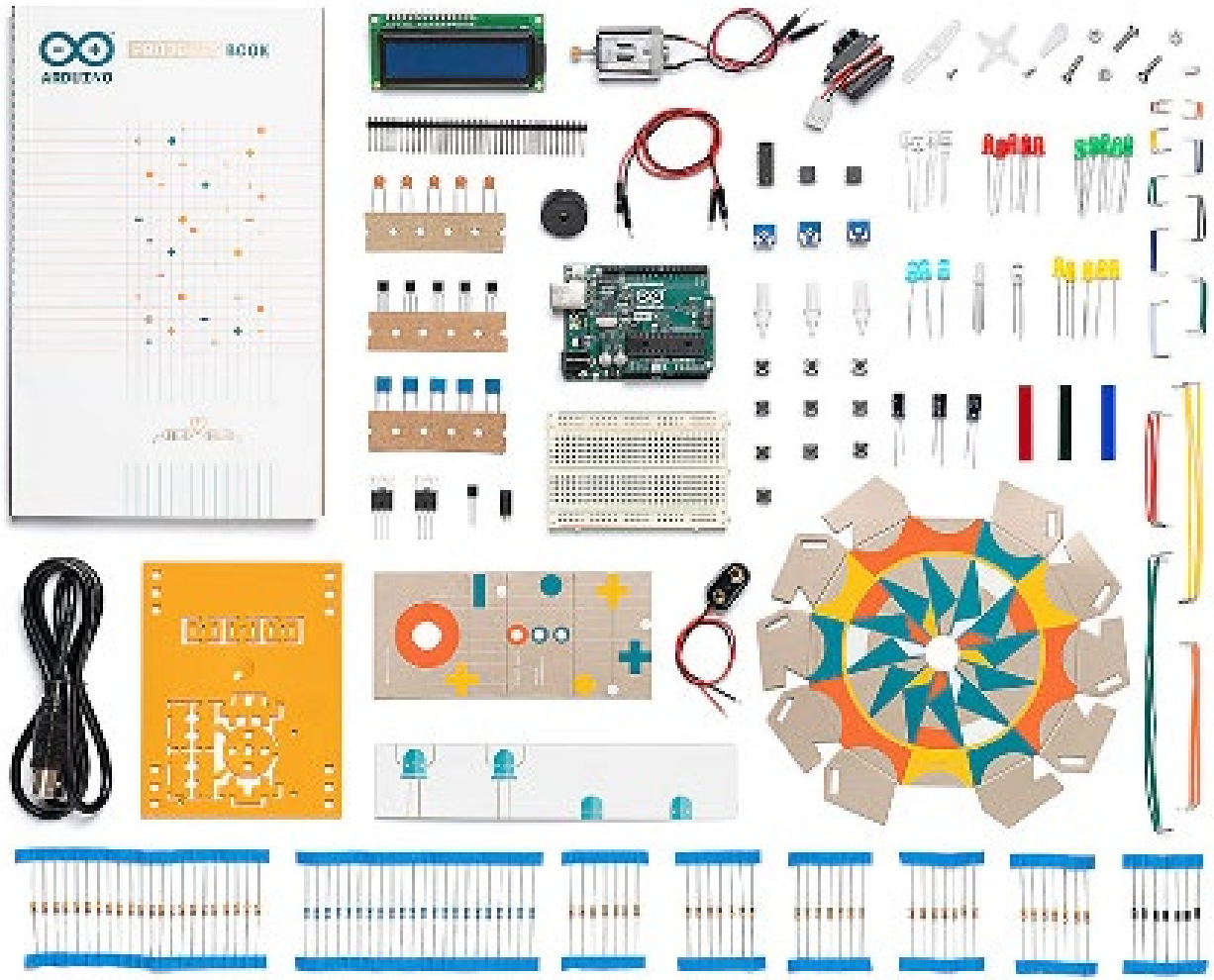
Videos for projects on YouTube: https://www.youtube.com/playlist?list=PLT6rF_I5kknMU7GQ8yzqpYzr-NbC0fS_4

The projects covered the basics of the Arduino Programming Language and the use of the Arduino UNO R3 hardware.



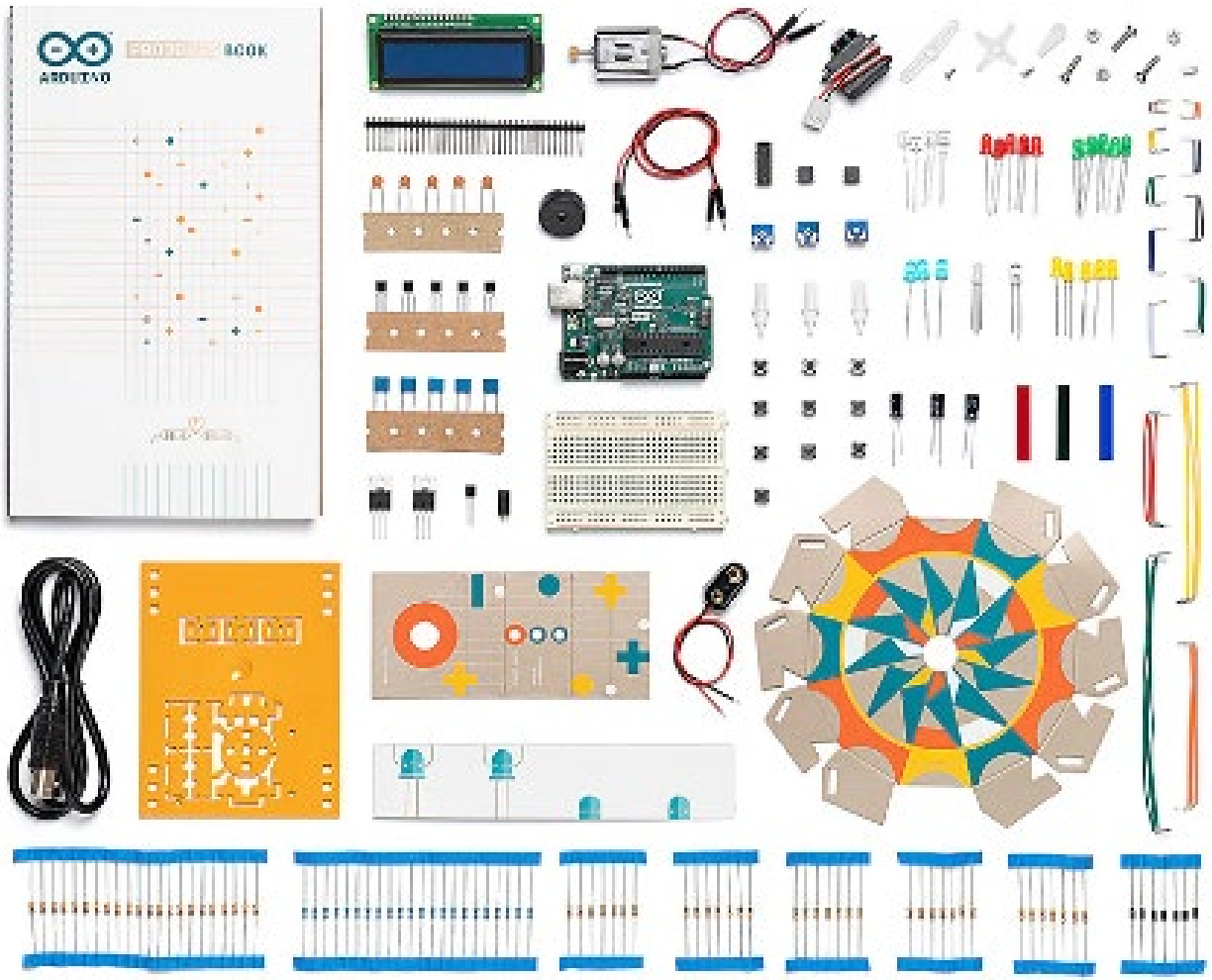
Phase 1a: Arduino Starter Kit, Parts list

Projects Book (170 pages)
Arduino Uno
USB cable
Breadboard 400 points
Solid core jumper wires (70)
Easy-to-assemble base
9v battery snap
Stranded jumper wire (black)
Stranded jumper wire (red)
Phototransistor (6)
Potentiometer 10kOhms (3)
Pushbuttons (10)
Temperature sensor [TMP36]
Tilt sensor
Alphanumeric LCD (16x2 characters)



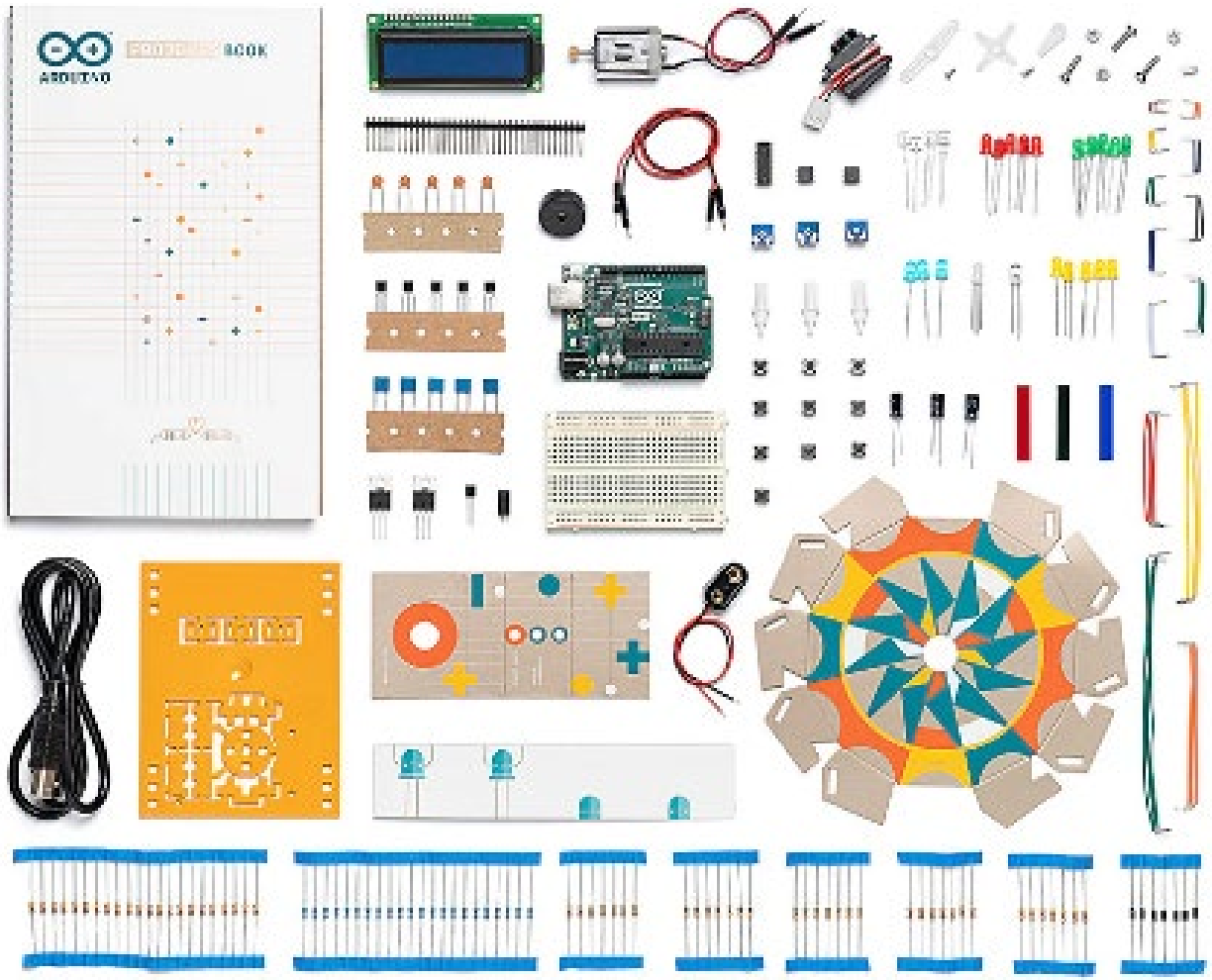
Phase 1a: Arduino Starter Kit, Parts list (continued)

LED (bright white)
LED (RGB)
LEDs (red) (8)
LEDs (green) (8)
LEDs (yellow) (8)
LEDs (blue) (3) (8)
Small DC motor 6/9V
Small servo motor
Piezo capsule [PKM22EPP-40]
H-bridge motor driver [L293D]
Optocouplers [4N35]
Mosfet transistors [IRF520] (2)
Capacitors 100 µF (3)
Diodes [1N4007] (5)
Transparent gels (red, green, blue)



Phase 1a: Arduino Starter Kit, Parts list (continued)

Male pins strip (40x1)
Resistors 220 Ω (20)
Resistors 560 Ω (5)
Resistors 1 kΩ (5)
Resistors 4.7 kΩ (5)
Resistors 10Ω (20)
Resistors 1 MΩ (5)
Resistors 10 MΩ (5)



Phase 1a: Arduino Starter Kit Projects

- 01 GET TO KNOW YOUR TOOLS an introduction to the basics
- 02 SPACESHIP INTERFACE design the control panel for your starship
- 03 LOVE-O-METER measure how hot-blooded you are
- 04 COLOR MIXING LAMP produce any color with a lamp that uses light as an input
- 05 MOOD CUE clue people in to how you're doing
- 06 LIGHT THEREMIN create a musical instrument you play by waving your hands
- 07 KEYBOARD INSTRUMENT play music and make some noise with this keyboard
- 08 DIGITAL HOURGLASS a light-up hourglass that can stop you from working too much

Phase 1a: Arduino Starter Kit, Projects (continued)

- 09 MOTORIZED PINWHEEL a colored wheel that will make your head spin
- 10 ZOETROPE create a mechanical animation you can play forward or reverse
- 11 CRYSTAL BALL a mystical tour to answer all your tough questions
- 12 KNOCK LOCK tap out the secret code to open the door
- 13 TOUCHY-FEEL LAMP a lamp that responds to your touch
- 14 TWEAK THE ARDUINO LOGO control your personal computer from your Arduino
- 15 HACKING BUTTONS create a master control for all your devices!

Phase 1a: Arduino Starter Kit, Evaluation

- With the exception of project 14, the projects in the book accompanying the Arduino Starter Kit were built and programmed. (Note: I didn't try to use the paper cut-outs)
- The projects gave a good introduction and were sufficient to help successfully pass the Arduino Certification exam.
- The Starter Kit also appeared to be a good introduction to electronic components, although the coverage of electronics theory had errors and could be improved
- The tilt sensor did not fit into the breadboard well and would fall out
- There appeared to be a part substitution for the "phototransistor" (datasheet given for photoresistor) that did not work well
- Concerns with safety with Project 15 which encouraged disassembling an electric fan to use Arduino to control it

Phase 1b: Arduino Certification

https://www.arduino.cc/education/certification

Access to Certification exam can be purchased with Arduino Starter Kit ($81.25 on 11/12/2025):

https://store-usa.arduino.cc/products/arduino-fundamentals-bundle

Available in different languages (I accidentally purchased Spanish version).

Online exam, 36 questions, 75 minutes to complete exam.

The exam is based on the concepts introduced in the Arduino Starter Kit. Can use the Starter Kit Projects book during the exam

Note that: “Candidates must be at least 16 years old to take the exam, and they will need parental or legal guardian consent if they are under 18, or in some countries if they are under 21. This depends on the age of consent in your country.”

Junior Certification and Arduino Junior Certification Bundle available (Note: I haven’t tried this).

Phase 1b: Arduino Certification, Evaluation

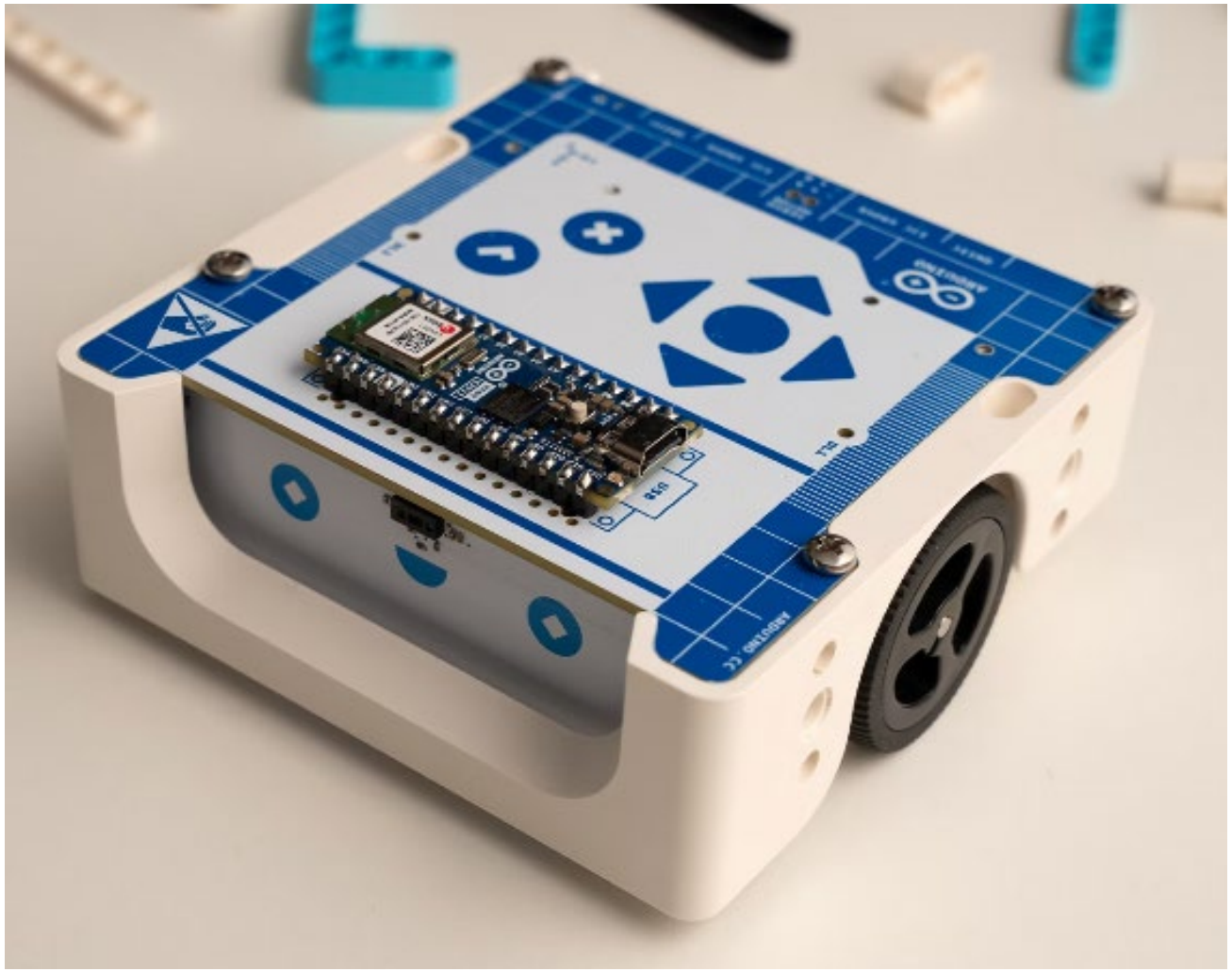
The projects in Arduino Starter Kit gave a good introduction and were sufficient to help me successfully pass the Arduino Certification exam.

I had one question whose answer I didn't find in the Starter Kit Projects book

Note: I didn't use the recommended browser when taking the certification exam. I used Microsoft Edge which did not show any figures.

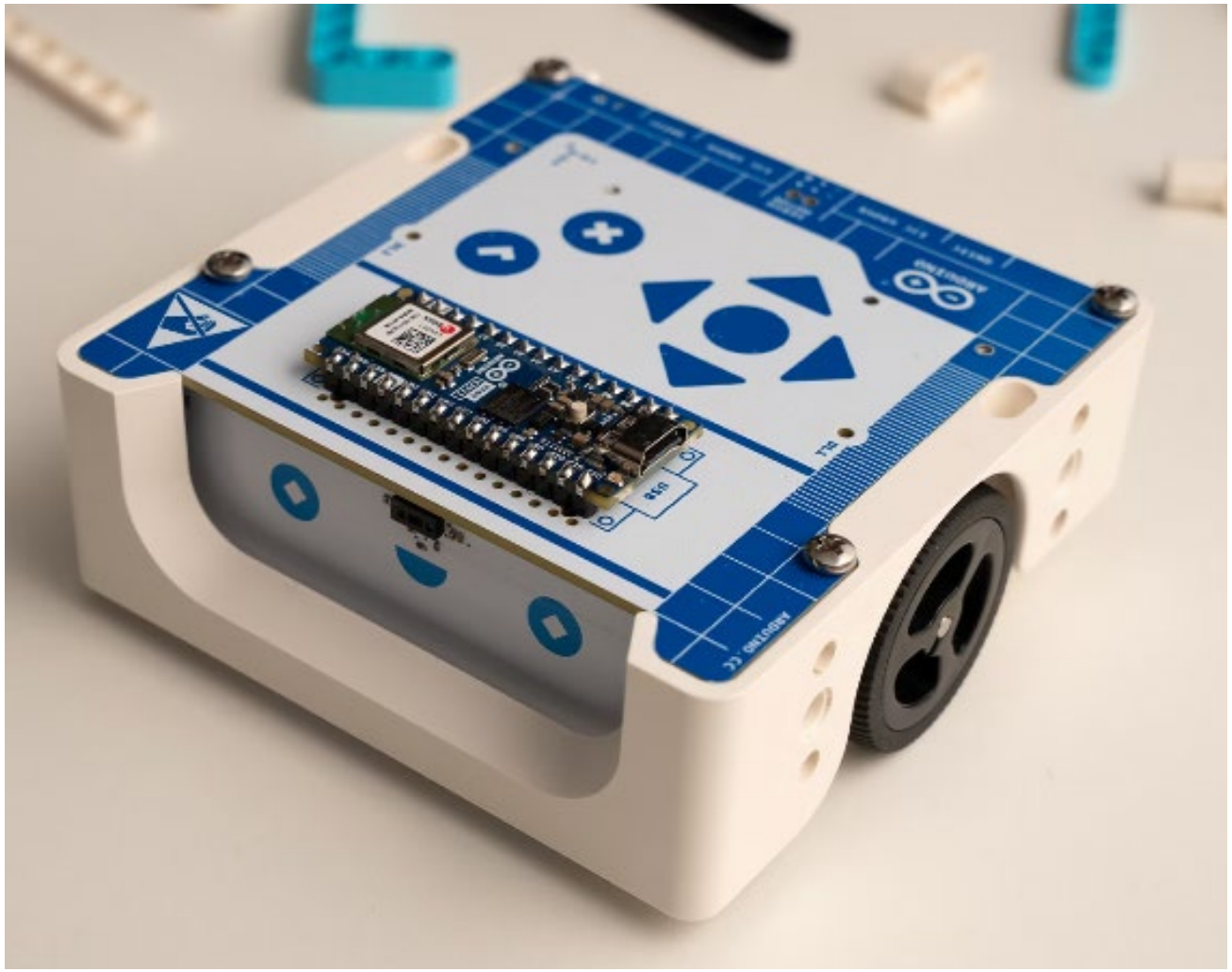
Phase 2
Arduino Alvik Robot

- Hypothesized that the Alvik robot could be used in the introductory EET 100 course to increase retention and also in recruitment activities such as a robotics workshop for high school students.
- The Alvik robots were promoted to be programmable by a variety of methods: the Arduino Programming Language, MicroPython, or graphical block-based coding.
- A set of Alvik robots was purchased



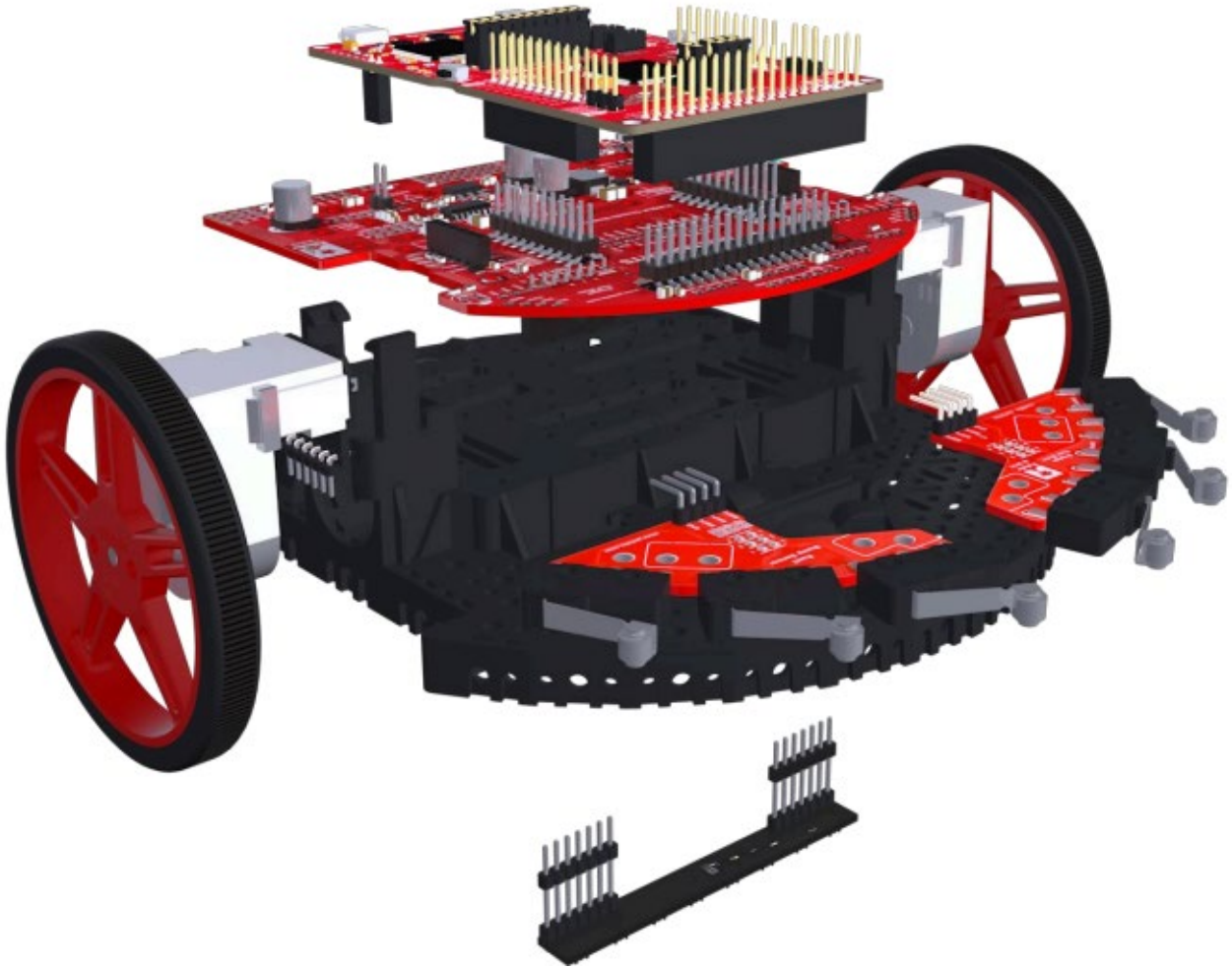
Phase 2
Arduino Alvik Robot, Evaluation

- Unfortunately, it was discovered that the Li-ion batteries in the robots were missing protective tabs, were deeply discharged, and needed to be replaced.
- As the robots were under warranty, replacement batteries were requested. Unfortunately, Arduino didn't supply a part number for replacement batteries. In addition, Arduino recommended batteries to be removed if robots aren't in use for long periods and the batteries were difficult to remove. Requested refund for Alvik robots.



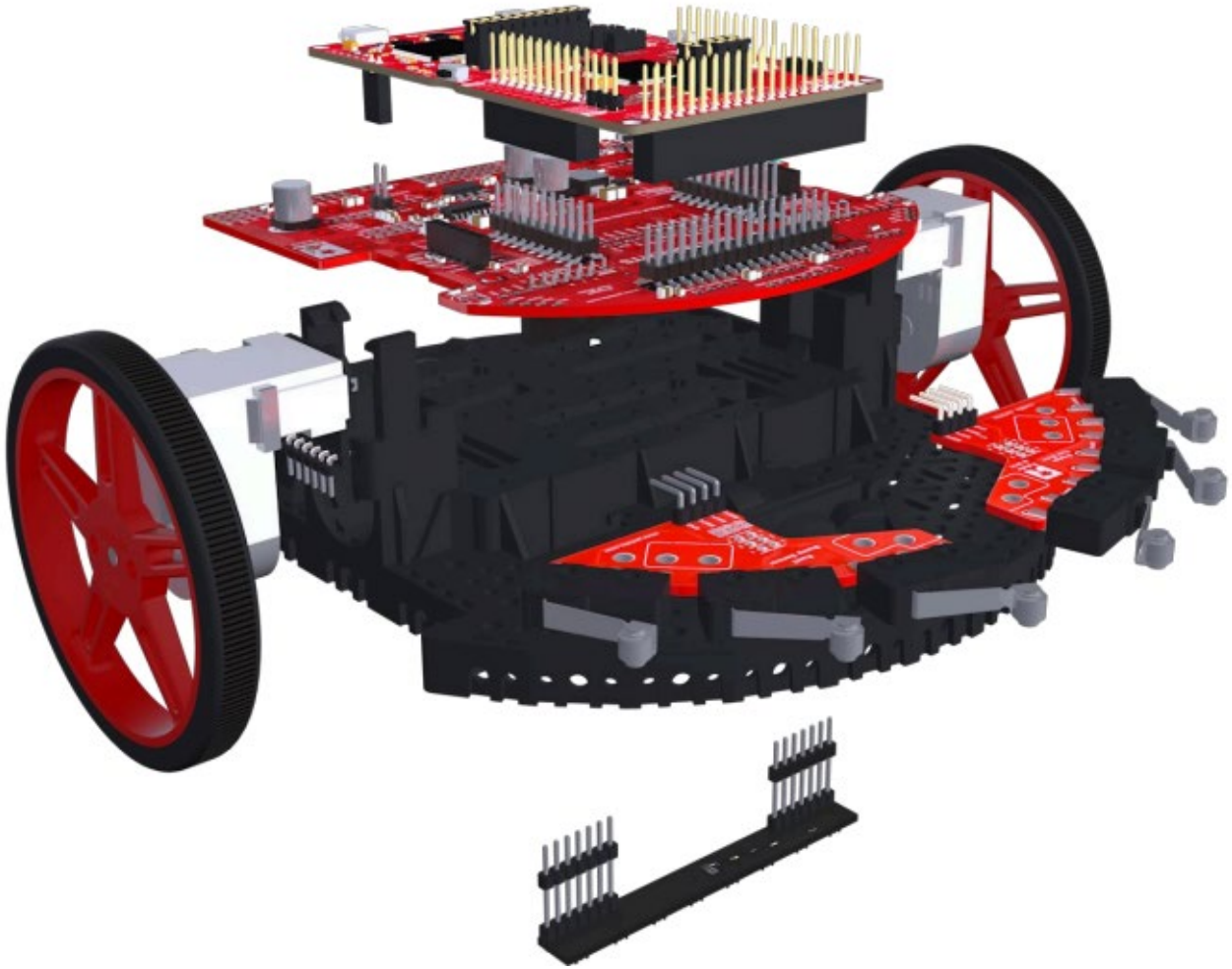
Phase 3
Retrofitting of TI-RSLK MAX robots to use Arduino hardware

- Approximately 50 TI-RSLK MAX Robotics System Learning Kits had been purchased for use in classes
- https://www.ti.com/lit/ml/sekp166/sekp166.pdf
- These kits were easy to assemble
- Used in EET 344 and 549 to create line-following robots and robots that would navigate using bumper switches to detect collisions.



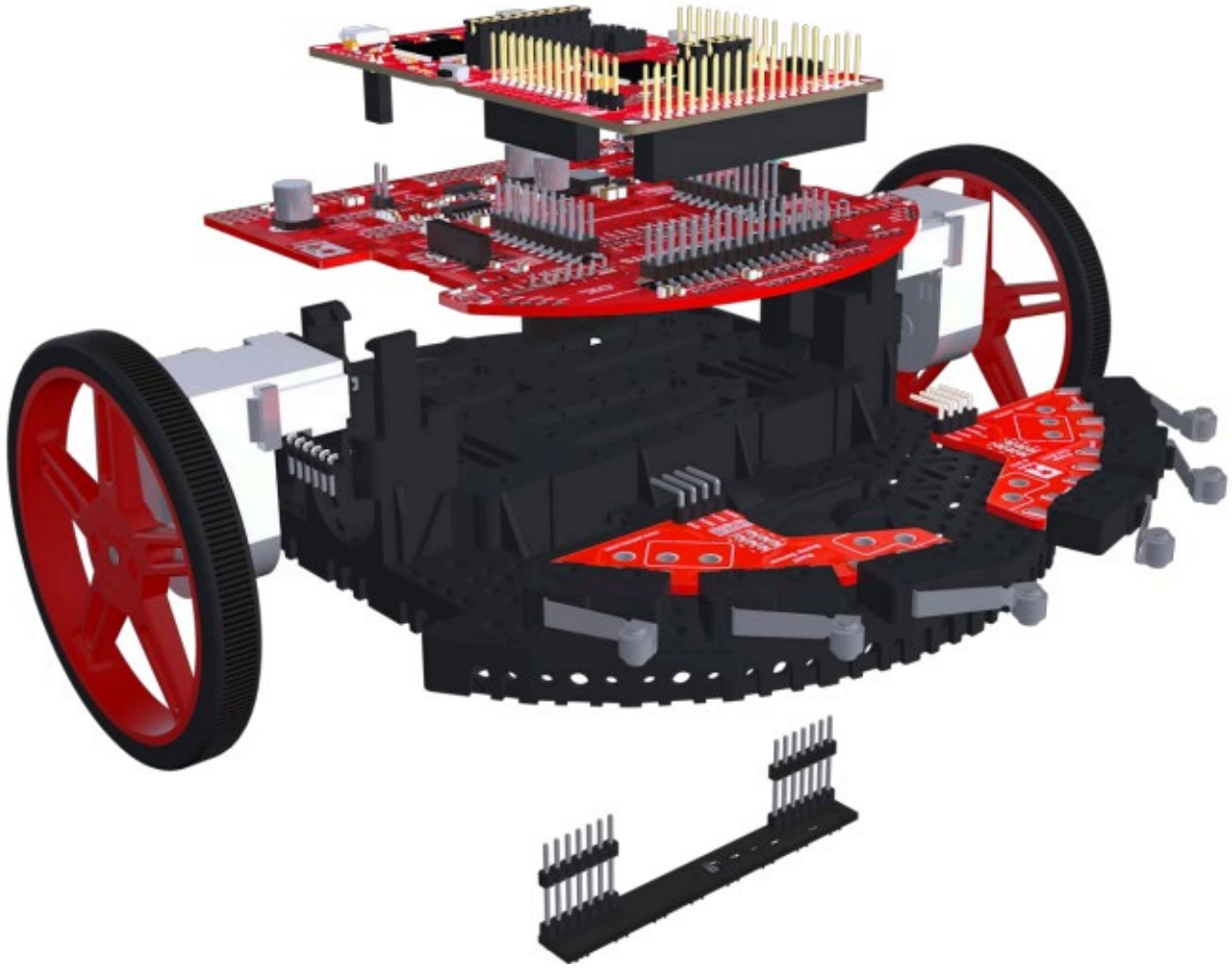
Phase 3
Retrofitting of TI-RSLK MAX robots to use Arduino hardware

- Later classes of EET 549 trained neural networks, a form of artificial intelligence (AI), to control the robot to follow a line which resulted in a conference paper [6]
- Unfortunately, TI stopped manufacturing the MSP432 microcontroller in the TI-RSLK MAX and stopped selling the robot kits.



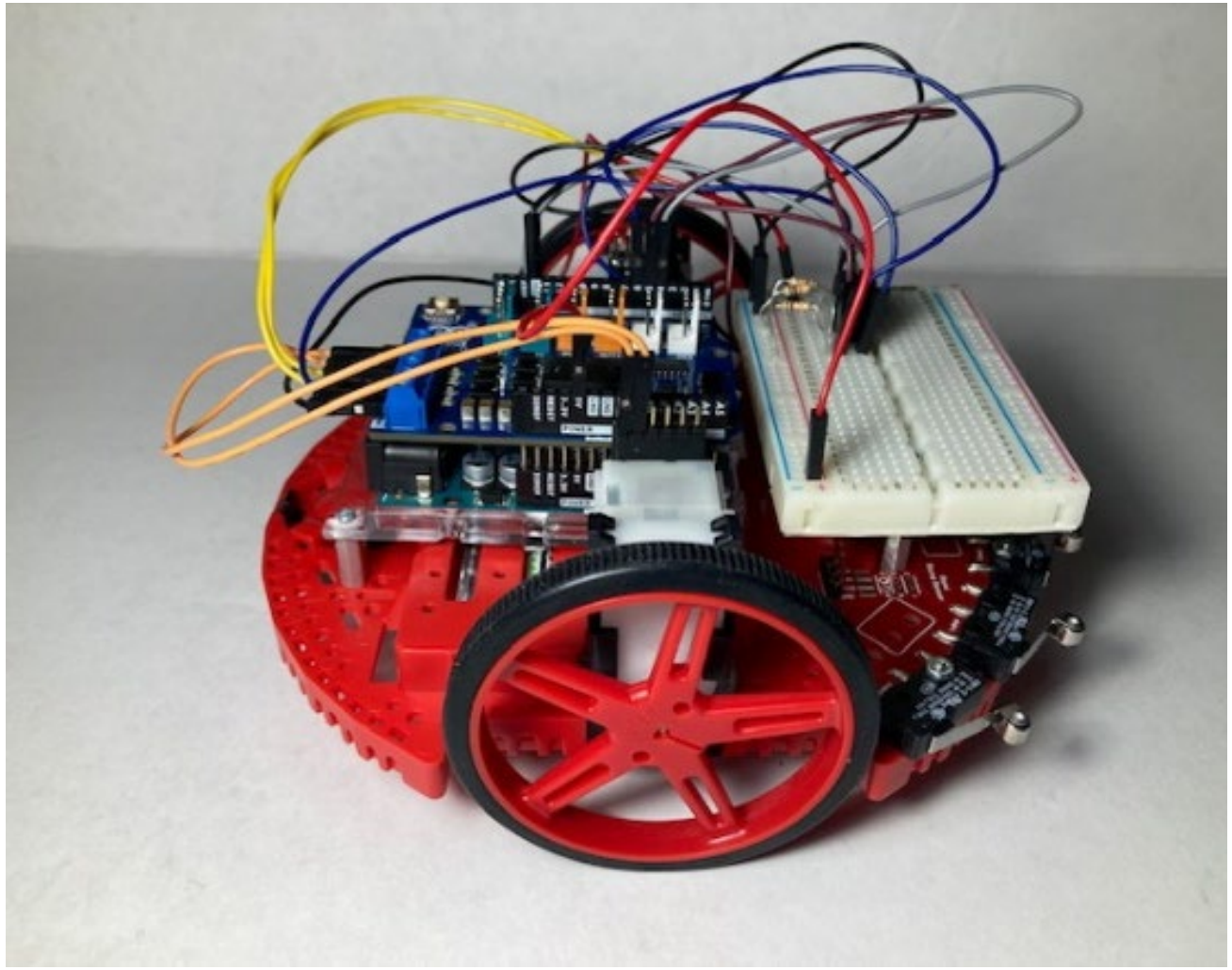
Phase 3
Retrofitting of TI-RSLK MAX robots to use Arduino hardware

- The majority of the parts for the TI-RSLK MAX, including the Romi chassis, were manufactured by Pololu and most are still available [7].
- Pololu makes a variety of other parts that fit the Romi chassis including power switches, motor drives, and Arduino-compatible microcontroller boards [8]-[10].
- Several boards were purchased from Pololu to evaluate for use in a retrofitting the robots.



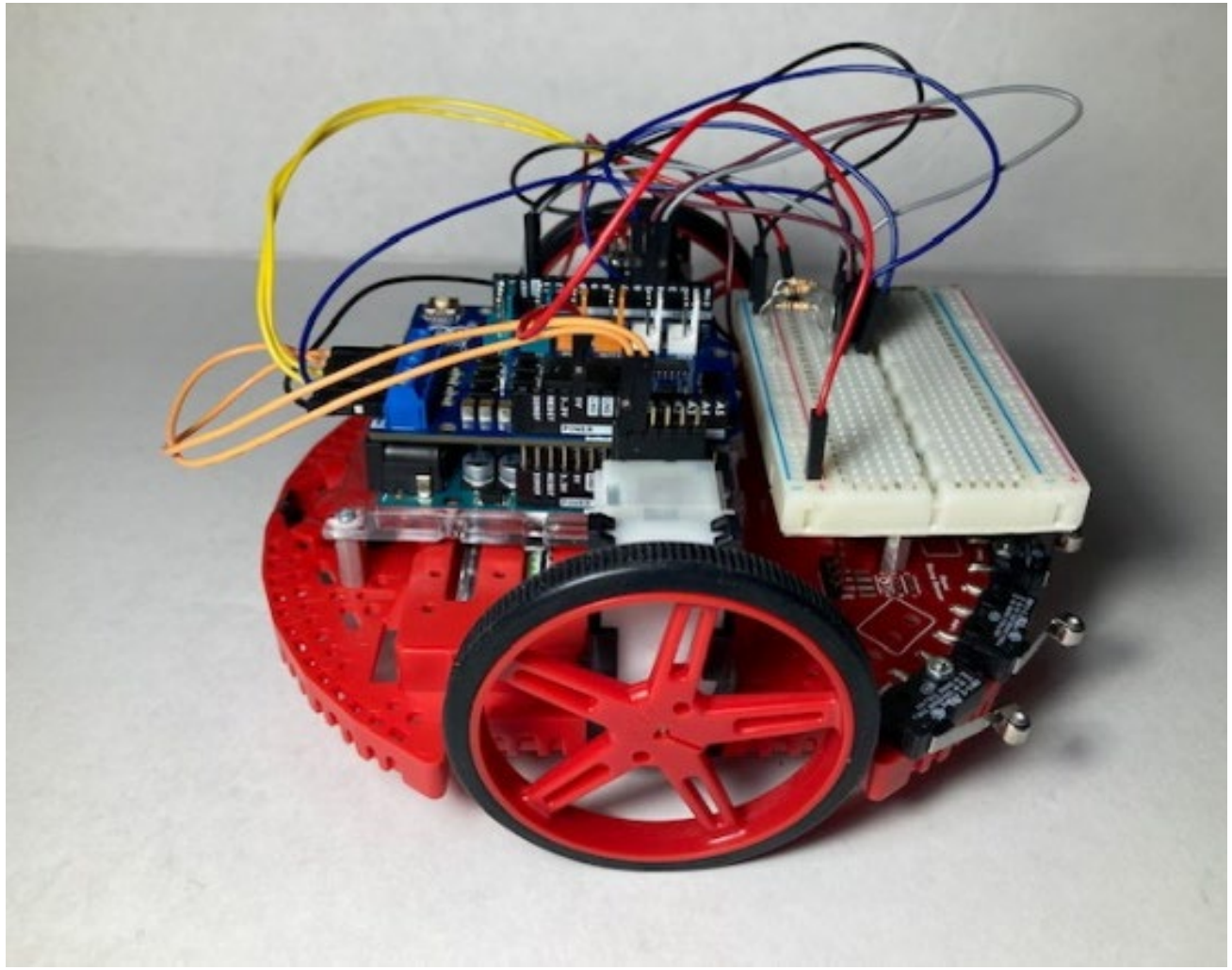
Phase 3
Retrofitting of TI-RSLK MAX robots to use Arduino hardware

- After looking at several replacement boards, it was decided that the best solution was to use Arduino UNO R3 boards and Arduino Motor Shields Rev3
  - Arduino UNO boards and motor shields could be used in other projects
  - Programming only required basic Arduino programming with basic instructions such as DigitalRead() and AnalogWrite() that were covered in Arduino Starter Kit with exception of configuring pull-up resistors in the microcontroller
  - Motor Shield had current measuring capability



Phase 3
Retrofitting of TI-RSLK MAX robots to use Arduino hardware

- The remainder of the parts could be reused from the TI-RSLK MAX kits with the exception of battery contacts, shorter standoffs, and minimal additional hardware.
  - Minimal soldering required to four battery contacts
- Will be testing 10 retrofitted robots for classroom use in EET 100 during Fall 2025
  - Planning to survey effectiveness of robots on student motivation to learn electronics



Pololu Romi Chassis Kit

Re-used chassis from TI-RSLK MAX which is still available from Pololu

https://www.pololu.com/product/3502

(Part number for red, available in different colors)

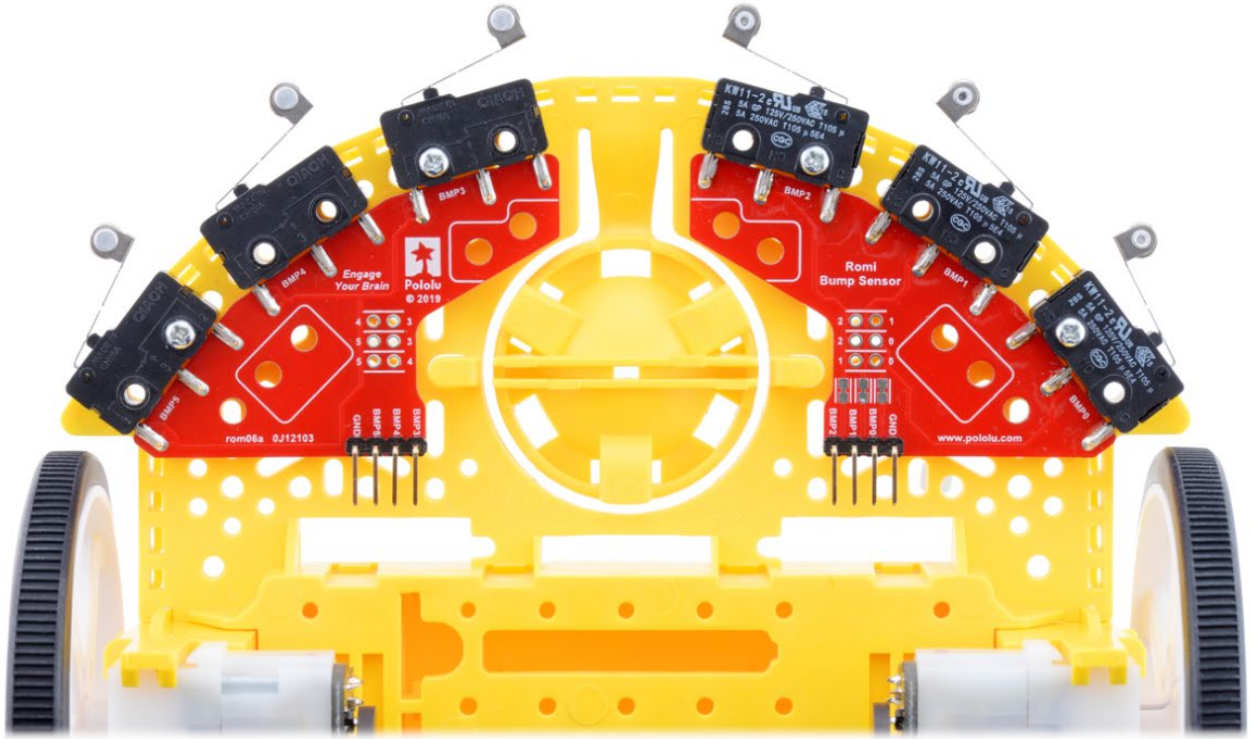
$39.95 (Price break at quantity 5, 25, and 100) on 11/13/2025

- One red Romi chassis base plate
- Two mini plastic gearmotors (120:1 HP with offset output and extended motor shaft)
- A pair of red Romi chassis motor clips
- A pair of white 70×8mm Pololu Wheels
- One red Romi chassis ball caster kit
- One Romi chassis battery contact set
  - Needed to purchase addition battery contact sets since some contacts where soldered to TI-RSLK MAX board

Bumper switches

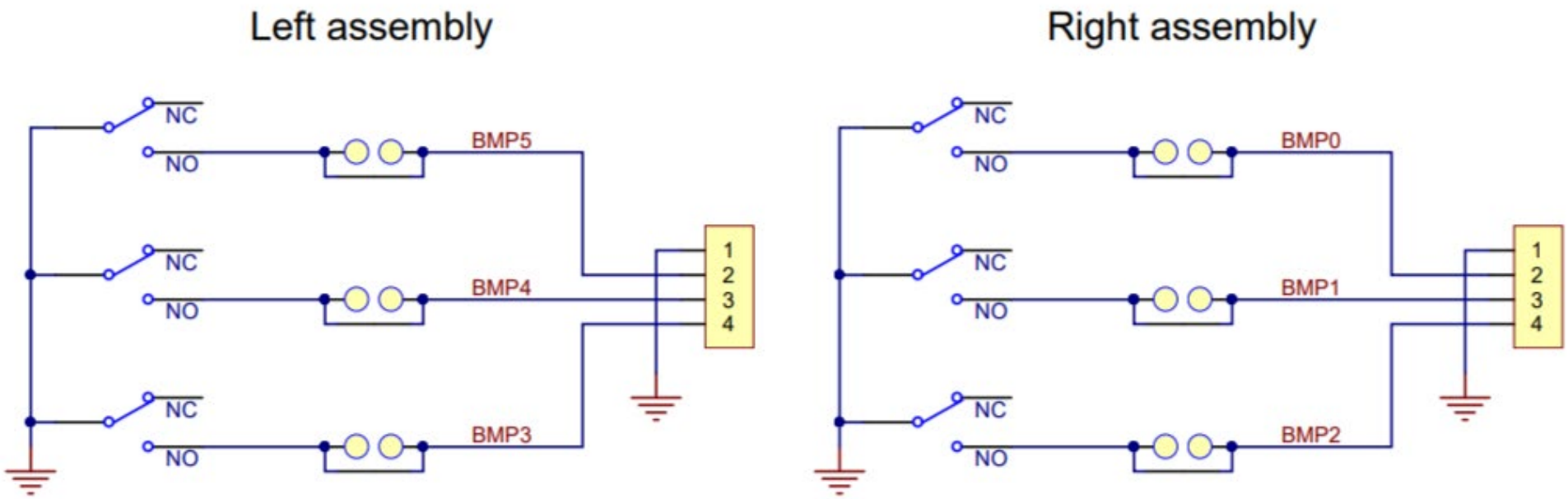
Re-used bumper switches from TI-RSLK MAX also still available from Pololu

- Left Bumper Switch Assembly for Romi/TI-RSLK MAX
  - $12.95 on 11/13/2025
  - https://www.pololu.com/product/3673
- Right Bumper Switch Assembly for Romi/TI-RSLK MAX
  - $12.95 on 11/13/2025
  - https://www.pololu.com/product/3674



Bumper switches schematic

- When switches closed, BMPx will be low.
- To make BMPx high when switches are open, need pull-up resistors connecting BMPx to 5 V (3.3 V? I need to verify this)
  - ATmega 328P in Arduino UNO R3 has interior pull-up resistors



Arduino UNO R3

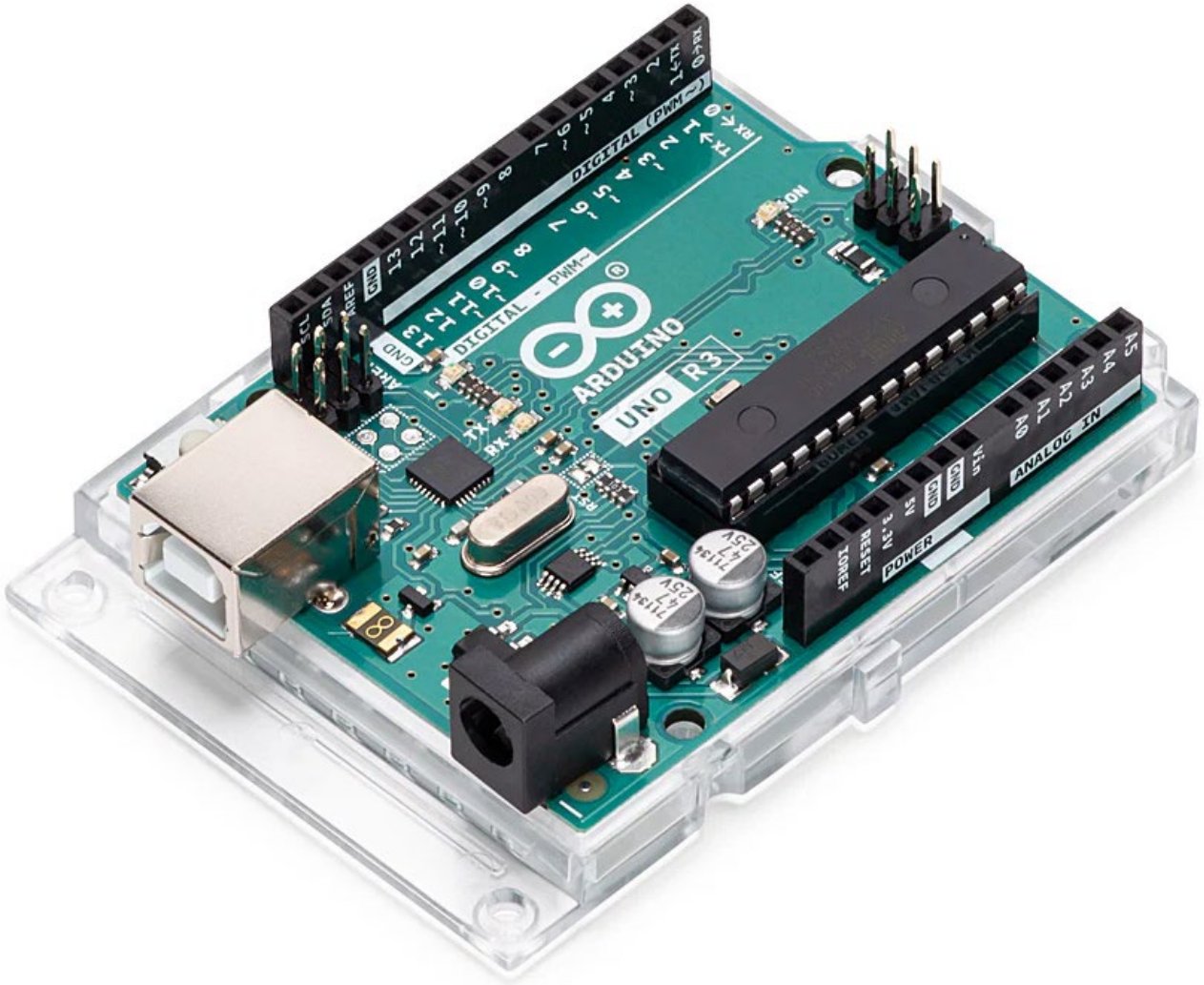
https://store-usa.arduino.cc/products/arduino-uno-rev3

$17.94 on 11/12/2025

https://docs.arduino.cc/hardware/uno-rev3/

Uses ATmega328P microcontroller (small, self-contained computer)

- 32 kB Flash (for programs)
- 2 kB SRAM (for variables)



Arduino Motor Shield Rev 3

https://docs.arduino.cc/hardware/motor-shield-rev3/

$28.40 on 11/13/2025

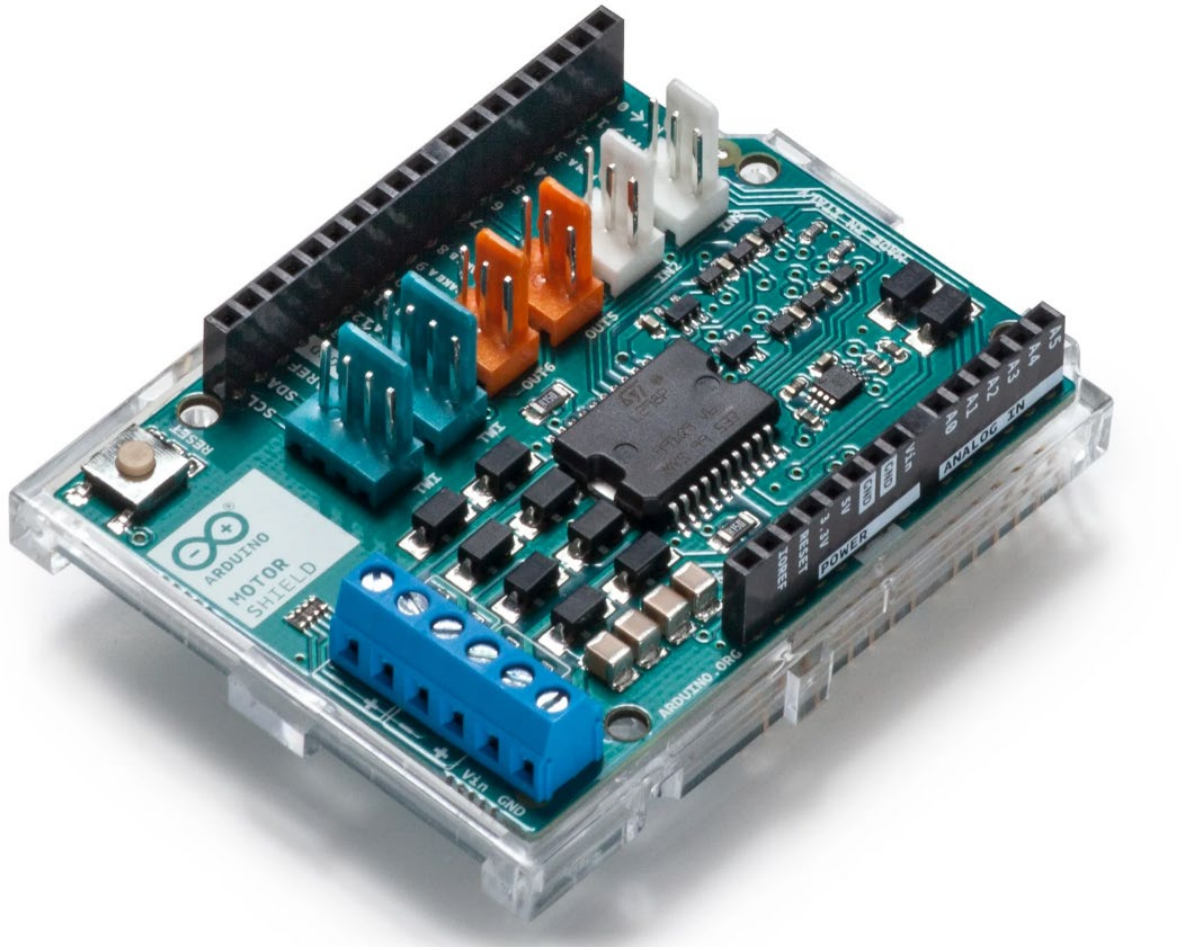
https://store-usa.arduino.cc/products/arduino-motor-shield-rev3

Operating Voltage: 5V to 12V

Motor controller: L298P, drives 2 DC motors or 1 stepper motor

Max current: 2A per channel or 4A max (with external power supply)

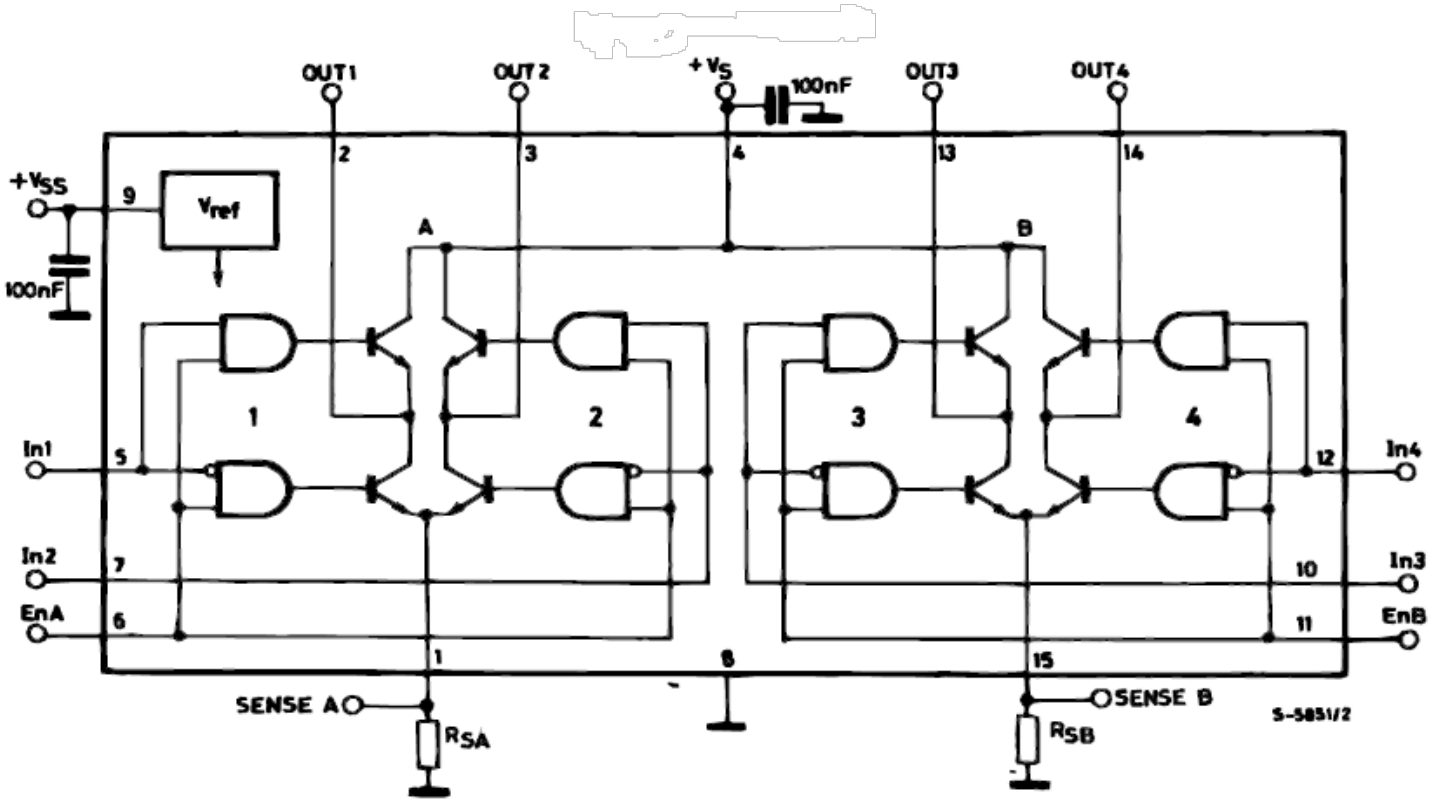
Current sensing: 1.65V/A



L298P Dual full-bridge driver

https://www.st.com/resource/en/datasheet/l298.pdf

- Operating supply voltage up to 46 V.
- Total dc current up to 4 A.
- Overtemperature protection.



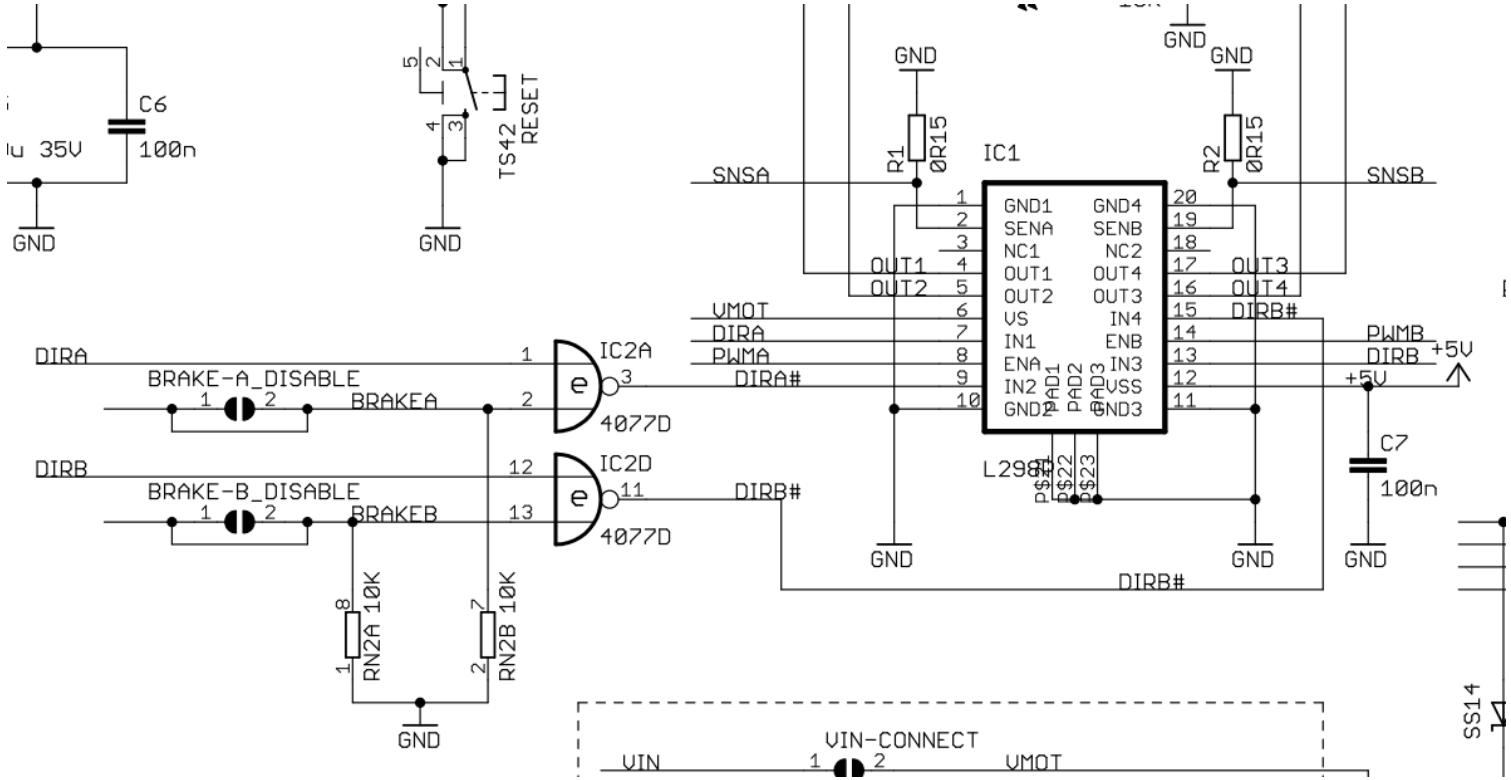
Arduino Motor Shield Rev 3 schematic, Connections to L298P

https://docs.arduino.cc/resources/schematics/A000079-schematics.pdf

- NAND gate: Output will only be 0 if both inputs are 1 (Is schematic correct?)

https://store-usa.arduino.cc/products/arduino-motor-shield-rev3

Function	pins per Ch. A	pins per Ch. B
Direction	D12	D13
PWM	D3	D11
Brake	D9	D8
Current Sensing	A0	A1



Arduino Motor Shield Rev 3, connections to batteries and dc motors

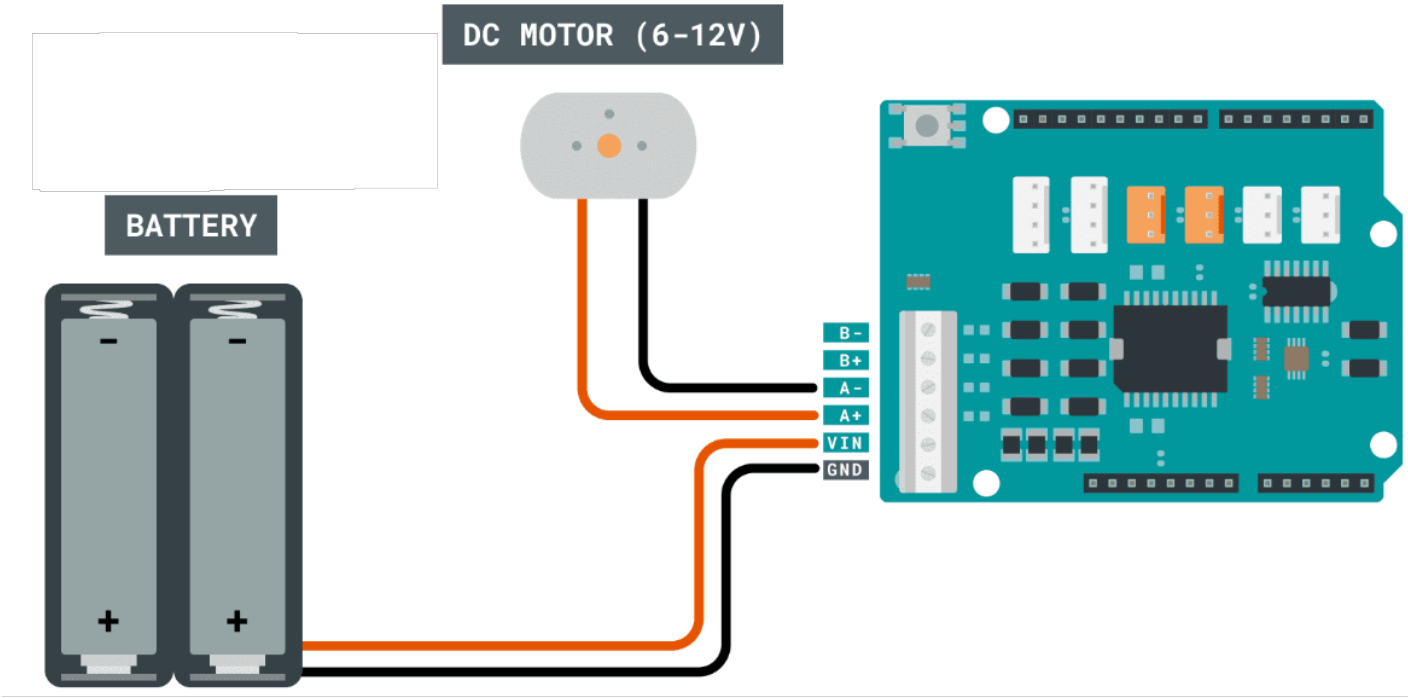
https://docs.arduino.cc/hardware/motor-shield-rev3/

Motor Shield connects with headers on top of Arduino UNO

https://docs.arduino.cc/tutorials/motor-shield-rev3/msr3-controlling-dc-motor/

Romi wired such that (looking from back of robot with bumper switches in front) Channel A is the left motor and Channel B is the right motor. See pinout and example code here

Note: I wouldn't recommend using Li-ion batteries.



Example Program

Program description

- Robot motors initially stopped
- Pressing any bumper switch BMP0 through BMP5 causes the robot to stop, reverse, stop, spin clockwise, stop, and then move forward.
- Pressing reset stops robot (program starts over)

Example Program

Use pinMode() to enable pull-up resistors in ATmega 328P microcontroller in Arduino UNO R3

- https://docs.arduino.cc/language-reference/en/functions/digital-io/pinMode/
- “It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-up.”

Use analogWrite() for Pulse-width modulation (PWM) for controlling motor speed

- https://docs.arduino.cc/language-reference/en/functions/analog-io/analogWrite/
- 500 Hz? (I need to verify frequency) PWM that uses range 0-255

Use digitalRead() to determine if bumper switches are open or closed

- https://docs.arduino.cc/language-reference/en/functions/digital-io/digitalread/

Example Program, initializing global variables for pin numbers for Motor Shield Rev 3

```
// Program: Robot initially off
// Pressing bumper switch BMP0 through BMP5 causes robot to backup and spin.
// Use reset to stop robot (program starts over)
// Note: pin 13 used for directionPinB is also pin for built-in LED LED_BUILTIN

// Pins for Channel A connected to left motor (looking from rear of robot)
int directionPinA = 12;
int pwmPinA = 3;
int brakePinA = 9;

//Pins for Channel B connected to right motor (looking from rear of robot)
int directionPinB = 13;
int pwmPinB = 11;
int brakePinB = 8;
```

Example Program, initializing global variables for pin numbers for Bumper switches

```
//Pins for right Bumper switches (looking from rear of robot)
int BMP0 = 5;
int BMP1 = 4;
int BMP2 = 2;

//Pins for left Bumper switches (looking from rear of robot)
int BMP3 = 6;
int BMP4 = 7;
int BMP5 = 10;
```

Example Program, setup(), set up pin directions as outputs for Motor Shield

```
void setup() { //setup() function only executed once

  //define pins
  pinMode(directionPinA, OUTPUT);
  pinMode(pwmPinA, OUTPUT);
  pinMode(brakePinA, OUTPUT);

  pinMode(directionPinB, OUTPUT);
  pinMode(pwmPinB, OUTPUT);
  pinMode(brakePinB, OUTPUT);
```

Example Program, setup(), set up pin directions as inputs and enable pull-up resistors for Bumper Switches (need schematic for switches?)

```
  // Bumper switches HIGH when open and LOW when closed. INPUT_PULLUP enables
pull-up resistor in microcontroller

  pinMode(BMP0, INPUT_PULLUP);
  pinMode(BMP1, INPUT_PULLUP);
  pinMode(BMP2, INPUT_PULLUP);
  pinMode(BMP3, INPUT_PULLUP);
  pinMode(BMP4, INPUT_PULLUP);
  pinMode(BMP5, INPUT_PULLUP);
```

Example Program, setup(), set up pins on Motor Shield to prepare to go forward, set thebrakes, and stop motors

```
  //Write a high state to the direction pins so ready to move forward
  //(Romi wired so bumper switches in front)
  digitalWrite(directionPinA, HIGH);
  digitalWrite(directionPinB, HIGH);

  //Activate brakes
  digitalWrite(brakePinA, HIGH);
  digitalWrite(brakePinB, HIGH);

  //Set duty cycle for the motors to 0 (motors not turning)
  //(duty cycle has range 0 to 255)
  analogWrite(pwmPinA, 0);
  analogWrite(pwmPinB, 0);

} // End of setup() function
```

Example Program, loop(), if any bumper switches have been closed, STOP…

```
  void loop() { //loop() function will continuously repeat

  //execute statements inside if() statement if any switches are closed

  if((digitalRead(BMP0)==LOW)||(digitalRead(BMP1)==LOW)||(digitalRead(BMP2)==LOW)|
|(digitalRead(BMP3)==LOW)||(digitalRead(BMP4)==LOW)||(digitalRead(BMP5)==LOW))
  {
    //STOP

    //activate brakes
    digitalWrite(brakePinA, HIGH);
    digitalWrite(brakePinB, HIGH);

    //set work duty for the motor to 0 (motors not turning)
    //(duty cycle has range 0 to 255)
    analogWrite(pwmPinA, 0);
    analogWrite(pwmPinB, 0);

    delay(1000); //Delay in milliseconds (ms). 1000 ms = 1 second
```

Example Program, loop(), if any bumper switches have been closed, STOP, REVERSE…

```
//REVERSE

//write a low state to the direction pins
digitalWrite(directionPinA, LOW);
digitalWrite(directionPinB, LOW);

//release brakes
digitalWrite(brakePinA, LOW);
digitalWrite(brakePinB, LOW);

//set duty cycle of 50 for the motor (duty cycle has range 0 to 255)
analogWrite(pwmPinA, 50);
analogWrite(pwmPinB, 50);

delay(500); //Delay in milliseconds (ms). 1000 ms = 1 second
```

Example Program, loop(), if any bumper switches have been closed, STOP, REVERSE, STOP,

```
//STOP

//activate brakes
digitalWrite(brakePinA, HIGH);
digitalWrite(brakePinB, HIGH);

//set duty cycle for the motor to 0 (motors not turning)
//(duty cycle has range 0 to 255)
analogWrite(pwmPinA, 0);
analogWrite(pwmPinB, 0);

delay(1000); //Delay in milliseconds (ms). 1000 ms = 1 second
```

Example Program, loop(), if any bumper switches have been closed, STOP, REVERSE, STOP, SPIN CLOCKWISE,…

```
//SPIN CLOCKWISE

//write a low to direction pin for right motor (reverse)
//and a high to direction pin for left motor (forward)
digitalWrite(directionPinA, HIGH);
digitalWrite(directionPinB, LOW);

//release brakes
digitalWrite(brakePinA, LOW);
digitalWrite(brakePinB, LOW);

//set duty cycle of 50 for the motor (duty cycle has range 0 to 255)
analogWrite(pwmPinA, 50);
analogWrite(pwmPinB, 50);

delay(660); //Adjusted this experimentally
```

Example Program, loop(), if any bumper switches have been closed, STOP, REVERSE, STOP, SPIN CLOCKWISE, STOP, …

```
//STOP

//activate brakes
digitalWrite(brakePinA, HIGH);
digitalWrite(brakePinB, HIGH);

//Set duty cycle for the motor to 0 (duty cycle has range 0 to 255)
analogWrite(pwmPinA, 0);
analogWrite(pwmPinB, 0);

delay(1000); //Delay in milliseconds (ms). 1000 ms = 1 second
```

Example Program, loop(), if any bumper switches have been closed, STOP, REVERSE, STOP, SPIN CLOCKWISE, STOP, and MOVE FORWARD!

```
    //FORWARD

    //Set direction pins high to go forward
    digitalWrite(directionPinA, HIGH);
    digitalWrite(directionPinB, HIGH);
    //}

    //release brakes
    digitalWrite(brakePinA, LOW);
    digitalWrite(brakePinB, LOW);

    //set duty cycle to 50 for the motor (duty cycle has range 0 to 255)
    analogWrite(pwmPinA, 50);
    analogWrite(pwmPinB, 50);
  } // end of if() statement

} // end of loop() function
```

Future possibilities

- Use motor encoders
- Investigate less expensive motor drivers from Pololu
- Teacher Workshop?
- Artificial Intelligence with camera
- Instead of Arduino UNO, use field programmable gate array (FPGA)

References

[1] Bring Your Projects to Life with Arduino Software, https://www.arduino.cc/en/software/

[2] Arduino Certification, https://www.arduino.cc/education/certification/

[3] Arduino Starter Kit, https://store.arduino.cc/products/arduino-starter-kit-multi-language

[4] Alvik Robot, https://www.arduino.cc/education/arduino-alvik/

[5] TI-RSLK MAX Texas Instruments Robotics System Learning Kit User guide, https://www.ti.com/lit/ml/sekp166/sekp166.pdf

[6] Alshammari, B., & Mayer, E., & Woods, Z., & Smith, A., & Hernandez, E., & Birk, K., & Allison, T., & Brennon, J., & Chase, C., & Kincheloe, B., & Lenharth, D., & Chandra, K. (2022, August), Introducing Deep Learning on Edge Devices Using A Line Follower Robot Paper presented at 2022 ASEE Annual Conference & Exposition, Minneapolis, MN., https://peer.asee.org/introducing-deep-learning-on-edge-devices-using-a-line-follower-robot

[7] Romi Chassis and Accessories, https://www.pololu.com/category/202/romi-chassis-and-accessories

[8] Power Distribution Board for Romi Chassis, https://www.pololu.com/product/3541

[9] Motor Driver and Power Distribution Board for Romi Chassis, https://www.pololu.com/product/3543

[10] Romi 32U4 Control Board, https://www.pololu.com/product/3544

[11] Microchip Studio, https://www.microchip.com/en-us/tools-resources/develop/microchip-studio

Acknowledgments

- The 2025 Summer Enhancement Grant was supported by the Langford endowment
- This material is based upon work supported by the National Aeronautics and Space Administration under Cooperative Agreement No. 80NSSC20M0109. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration nor of Wichita State University.
- Kansas Space Grant Consortium (KSGC) - NASA in Kansas