UNBOXING AND INSTALLATIONS

Get started with the Arduino Engineering Kit. Get to know the materials included in the kit, the tools, the content structure, best practices and instructions to install the tools you will use to build the projects.

1.1 About the Kit

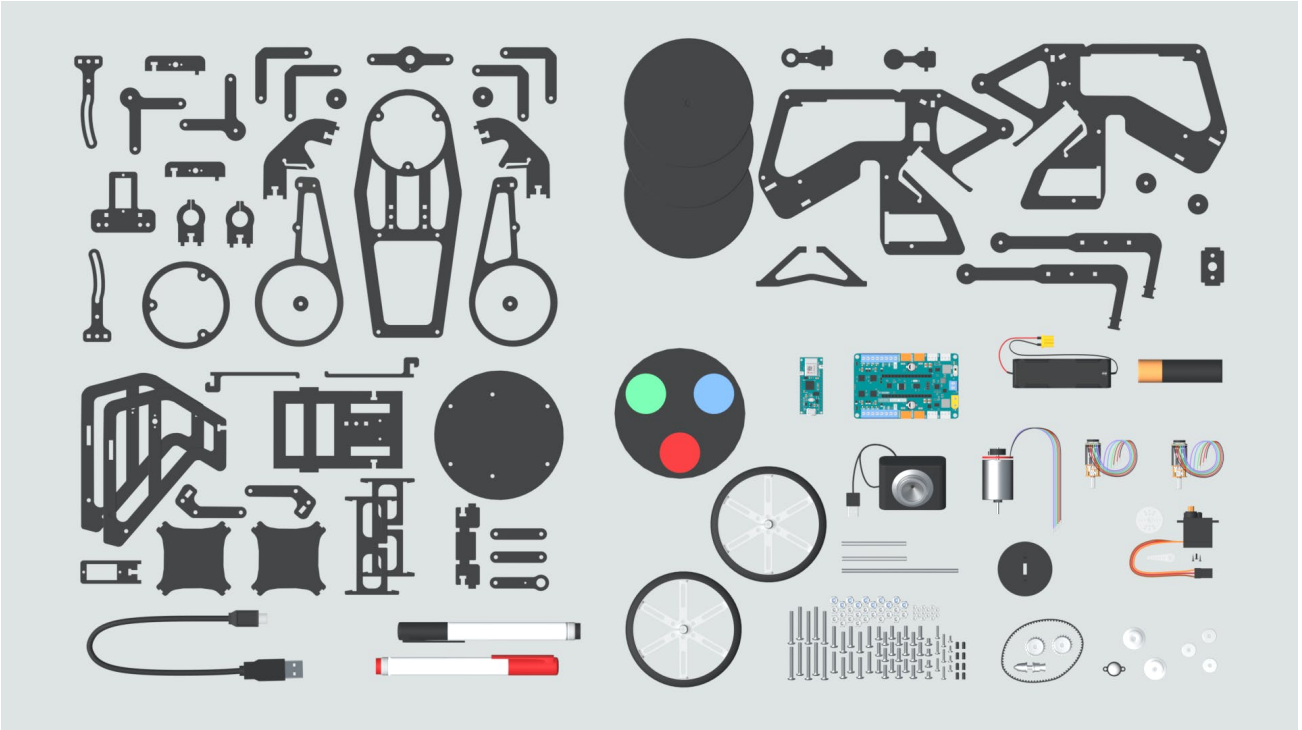
The Arduino Engineering Kit Rev2 comes with a set of electronic components, assembly pieces, software tools and educational materials that you will need to build the projects.

In this section you will learn:

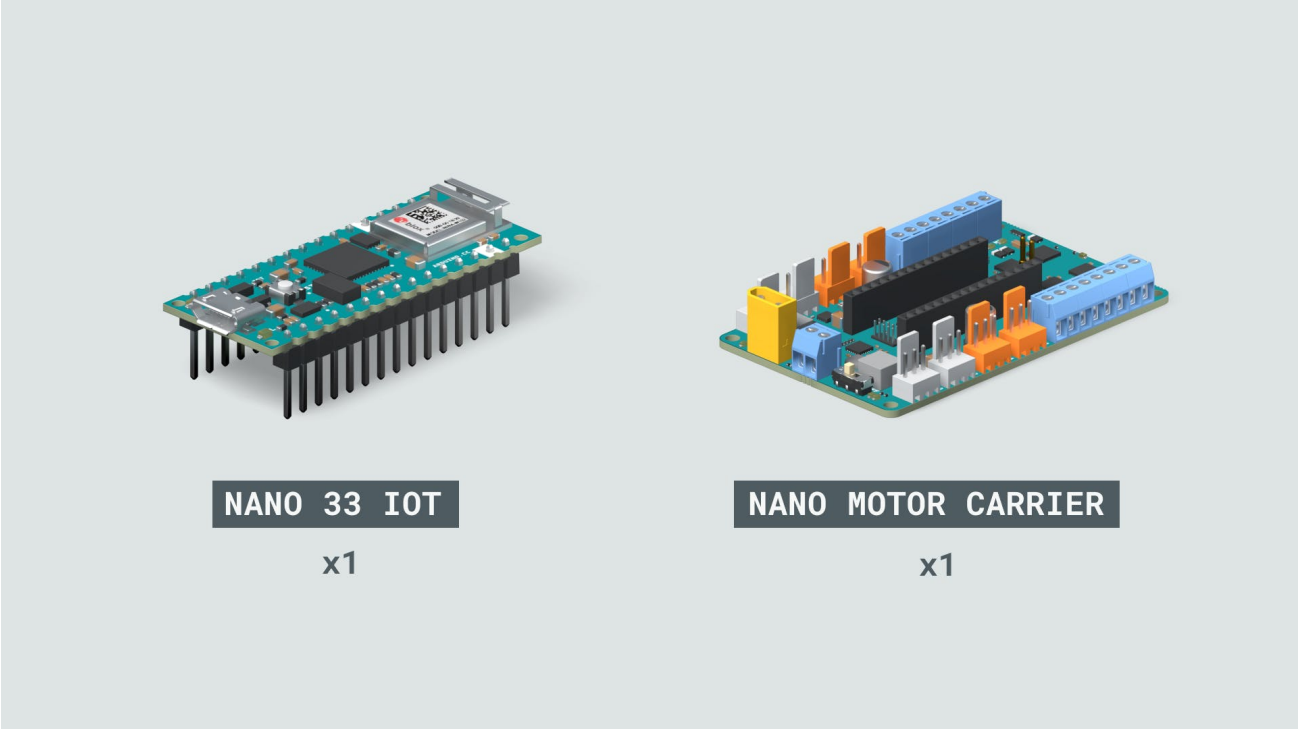
- About the various electronic components and other materials in the box,
- How the educational materials have been organised and their usage,
- About the purpose of the Arduino and MATLAB software tools.

In The Box

This section is a complete description of all the materials and components included in the kit, some of them you might be familiar with already, others will be completely new to you.



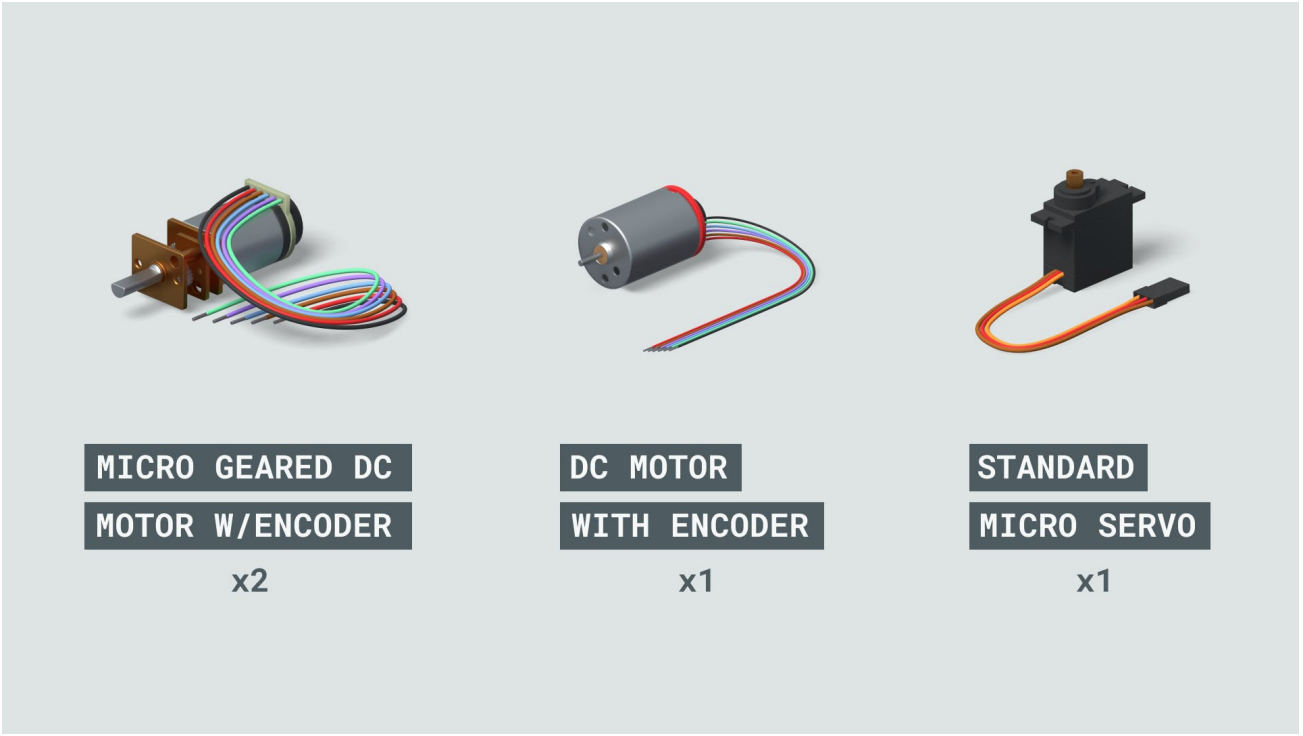
Arduino Board and Carrier



The Arduino Nano 33 IoT is an Arduino board based on the Atmel ATSAMW25 SoC. it features Wi-Fi connectivity and a small form factor. It is an effective board f concerning the Internet of Things due to its size and connectivity.

Arduino Nano Motor Carrier is an expansion board specially tailored for the Arduino Nano family of boards aimed at extending their functionality. This carrier enables the Arduino Nano 33 IoT to easily control servos, DC motors, and stepper motors.

DC and Servo Motors

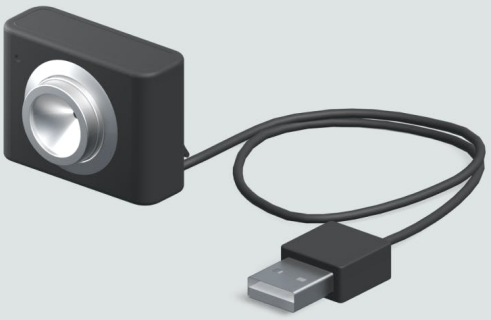


Micro Geared DC Motor with Encoders is a small 12V DC Motor with Encoder that features a 100:1 gearbox to increase the torque. We will talk more in detail about this and other types of motors in chapter 3.

Standard Micro Servo is a type of motor that can be controlled from an Arduino board without a need of any other components. Standard servos have a rotation angle limited to 180 degrees with a precise control over the angle of rotation.

DC Motor with Encoders is a larger 12 V DC motor that also features an encoder but does not include a gearbox. This motor will be used for spinning the inertia wheel on the motorcycle project.

Usb Web Cam



USB WEBCAM

x1

The Webcam is a USB connected digital camera that can record images by converting optical images to electrical signals in real-time. In our projects, it will be used for capturing images to be used in simple image recognition algorithms, and tracking movement of devices in control experiments.

Mechanical Parts

Caster wheel is a ball-shaped wheel that can roll in multiple directions which enables building dynamic mechanisms with using only three wheels.

Propellor adapter screw is a component that is used for mounting parts to a motor or a shaft securely. It prevents vibrations from occurring when the parts are spinning fast.

Metal shafts are used for transmitting rotation on a linear axis. The kit comes with a normal 1.90mm shaft and a 2.50mm D shaft. The D Shafts are used for transmitting rotation on a linear axis. The difference of the D shafts from the normal ones is the D-shaped cross section which creates a lock mechanism on the rotation axis.

Shaft collars are used for securing the positioning of the components on a shaft and prevents them from sliding over it.

Spools are like pulleys around which thread, wire or cable is wound, especially a cylinder or spindle.

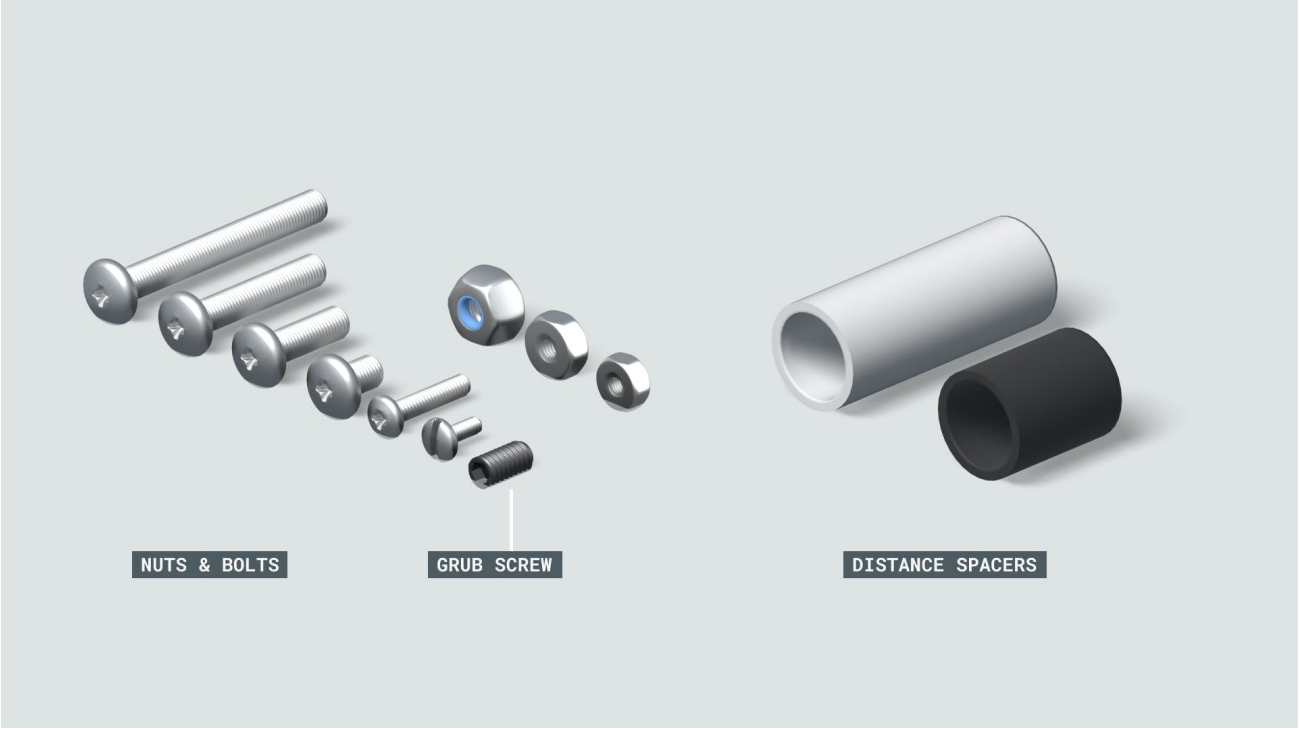
Timing belt and pulleys

The Timing Belt is an instrument that is used for transmitting motor motion to a wheel or a shaft. Timing belts have matching teeth with timing pulleys for

that no steps are missed while transmitting the motion between them.

A Timing Pulley is an instrument that is used for transmitting motor motion to a wheel or a shaft. Timing pulleys have matching teeth with the timing belt for ensuring that no steps are missed while transmitting the motion.

Nuts, Bolts and Distance Spacers

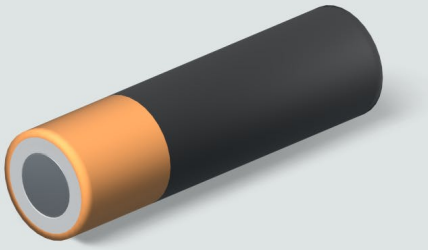


Nuts and bolts come in various sizes, together with assembly pieces, will be used for building the projects in chapter 4, 5, and 6.

Grub screw is a special screw is used to tighten the timing pulley, shaft collars and also the spool to the respective metal shafts.

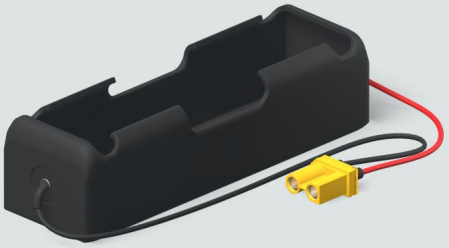
Distance spacers are plastic tubes which enable creating a distance between assembled pieces when needed.

LI-ION Battery and Holder



LI-ION BATTERY

x1

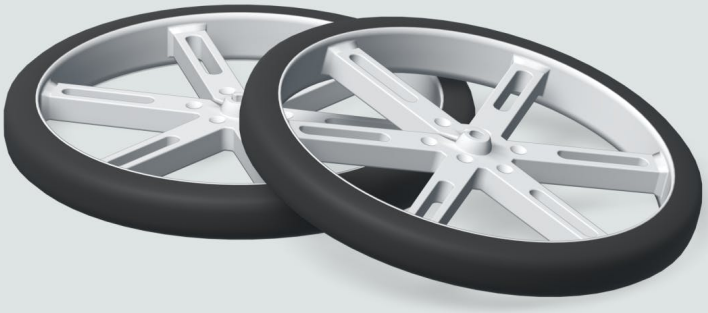


BATTERY HOLDER

x1

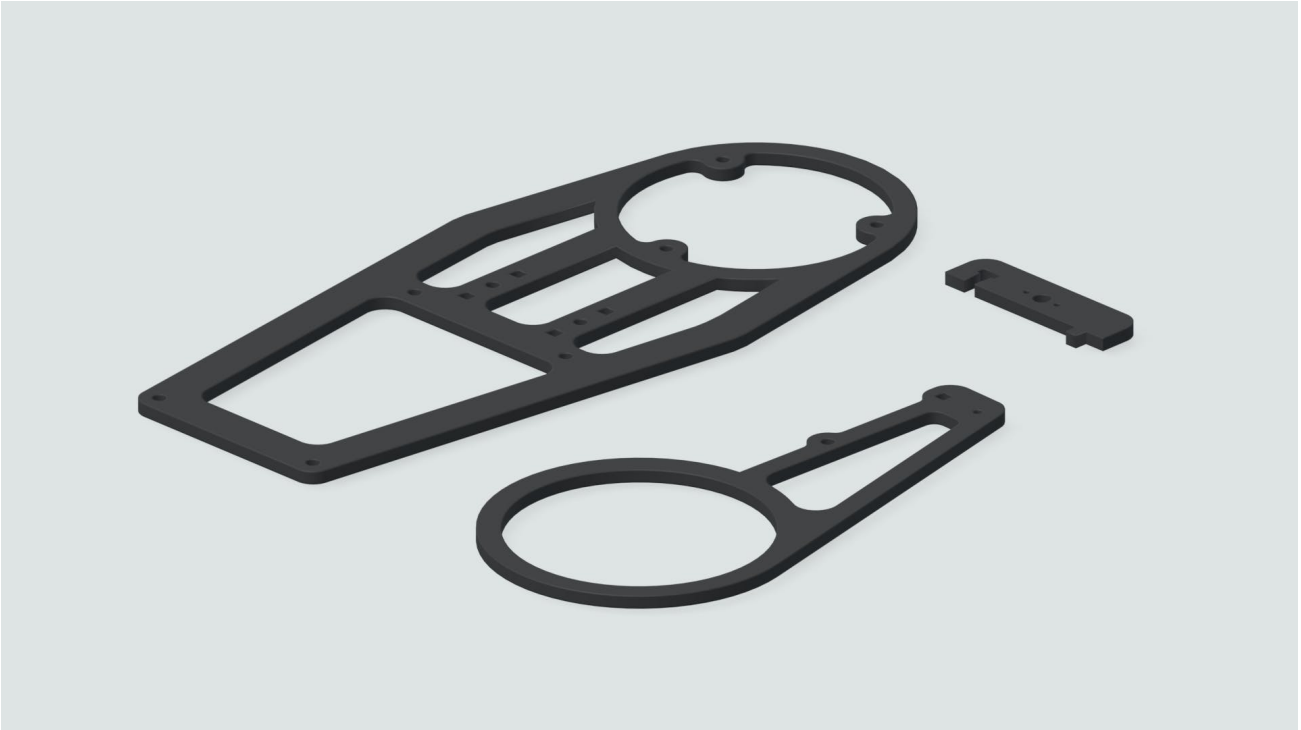
Batteries can power electronic circuits without connecting them to a computer. Using the battery holders you can attach the batteries to your projects in a safe manner.

Wheels



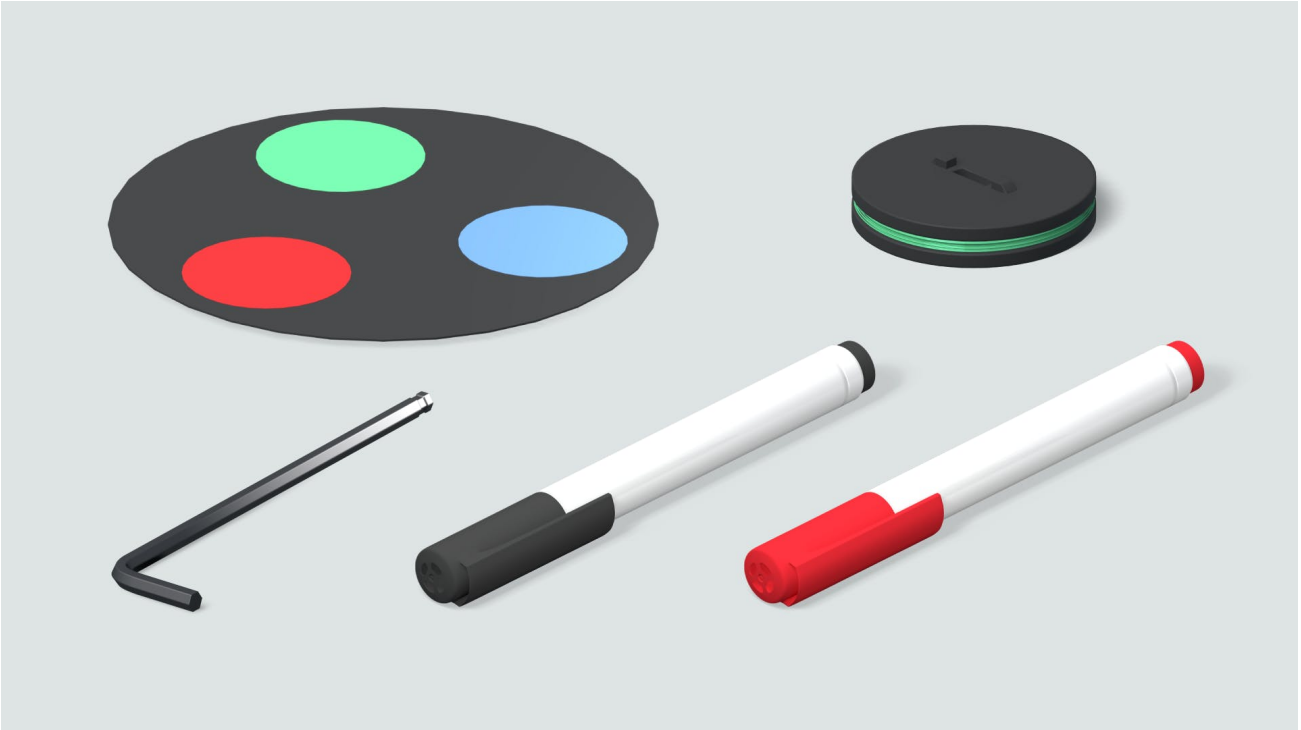
The wheels are used for moving the objects when spinning the motors. They will be used for the projects: Webcam Controlled Rover and Self Balancing Motorcycle.

Assembly Pieces



In order to build the projects in chapter 4, 5 and 6, you will use laser-cut ABS. They are stored in dedicated compartments boxes per project.

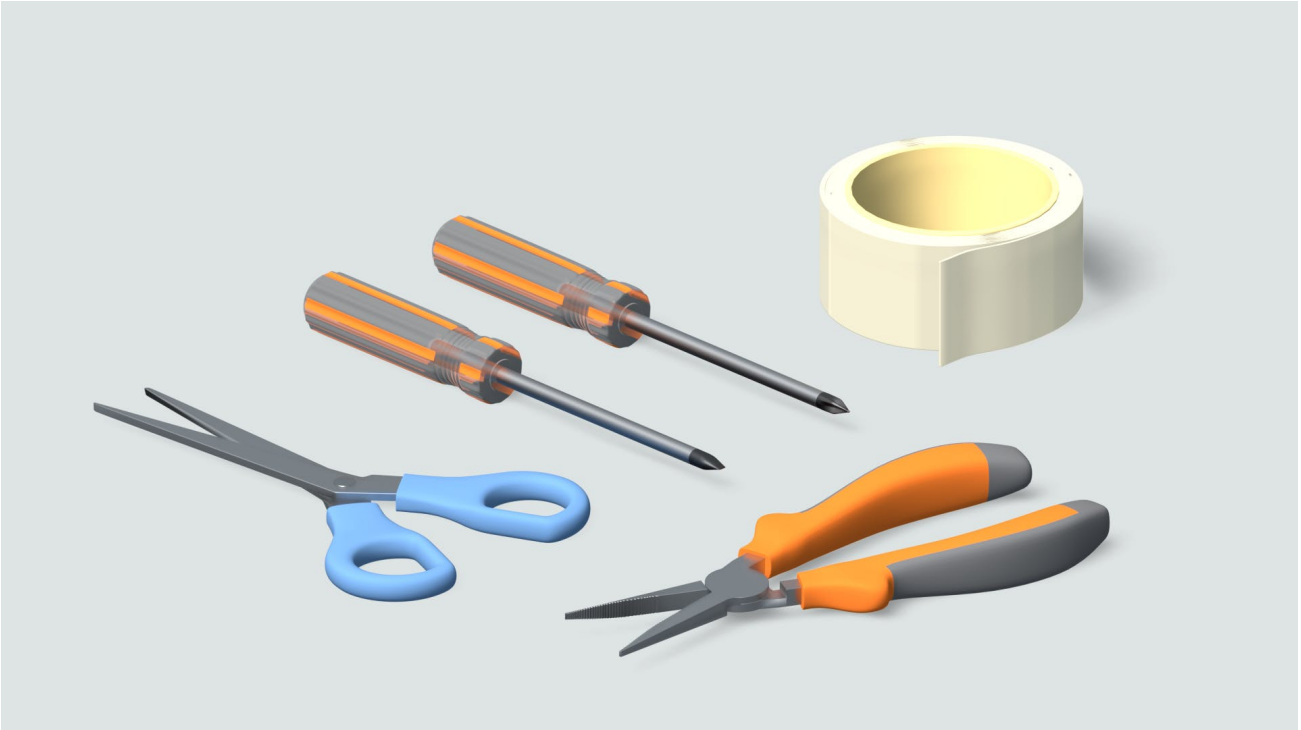
Additional Components



In addition to the components described, you will also find:

- Whiteboard drawing pens
- 5m thread
- Disk sticker of the Rover project
- Allen key

Components Needed (Not included in the Kit)



In addition to the components included in the kit, you will also need:

- Philips screwdriver
- Flat head screwdriver
- Scissors
- Tape
- Pliers
- Whiteboard

Note: A Philips #1 screwdriver and 2.5mm flat head screwdriver is recommended

Online Platform

All the content is available online, through the Arduino Education Online Pla[illegible]. The content for this kit is divided into 3 Lessons and 3 Projects:

Chapters 1, 2 and 3 are lessons, that focuses on

- Setting up the necessary tools and workspace,
- Acquiring a fundamental knowhow on the use of Arduino and MATLAB,
- Teaching critical programming and electronic concepts in relation to the components included in the kit,
- Experimenting, observing and evaluating important mechatronic concepts through MATLAB tools.

Chapters 4, 5 and 6 are the projects, that focuses on

- Developing real world applications that uses certain core concept and important MATLAB and Arduino Tools,
- Gaining an understanding of important problem solving techniques and processes,
- Evaluate the design of the projects and produce the most effective design for meeting the specifications and the constraints,
- Acquiring a proficiency in using the MATLAB and Simulation software.

The content has been designed to be followed sequentially from chapters 1 to 3, which should be used as reference when needed. However the projects can be done at any order, but it is recommended to start with the project in chapter 4.

Software Tools

In the Arduino Engineering Kit Rev2, you will program the Arduino Nano 33 IoT primarily using MATLAB and Simulink.

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. **Arduino boards** can read inputs, amount of light on a sensor, a p
button, or the arrival of a Twitter message and turn this into an output, for

activating a motor, turning on an LED or publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do this use the **Arduino programming language** and the **Arduino Software (IDE)**.

Thanks to the Open Source philosophy that underlies the project, Arduino gained the support of a huge community of users that share projects, tutorials and suggestion on the dedicated channels such as the **Arduino forum** and the **Arduino Project Hub**.

MATLAB

MATLAB is a high-level programming language and development environment for engineering and scientific computing. It is optimized for numeric computation and includes thousands of math and engineering functions that can be used to analyze and visualize data, develop algorithms, and create mathematical models. Dozens of add-on products extend MATLAB by adding domain specific functionality in areas such as controls, signal processing, robotics, optimization, image processing, and many more.

MATLAB Support Package for Arduino enables you to communicate with Arduino hardware directly in MATLAB. You can interactively read data from a variety of sensors and peripheral devices, process the raw data into meaningful quantities, and actuate external devices such as lights and motors. Because MATLAB is a high level interpreted language, you can see results from I/O instructions immediately without having to compile your code.

Simulink

Simulink is a block diagram environment for modeling and simulating dynamic systems and developing algorithms that you can embed into Arduino and other hardware. Operations and functions are represented by Simulink blocks, while data is represented by signals which flow from one block to another. With Simulink you can incorporate signal processing, control design, state logic, and other advanced math and engineering routines into your Arduino projects automatically, and thus the traditional software development processes are done for you. You can just focus on engineering!

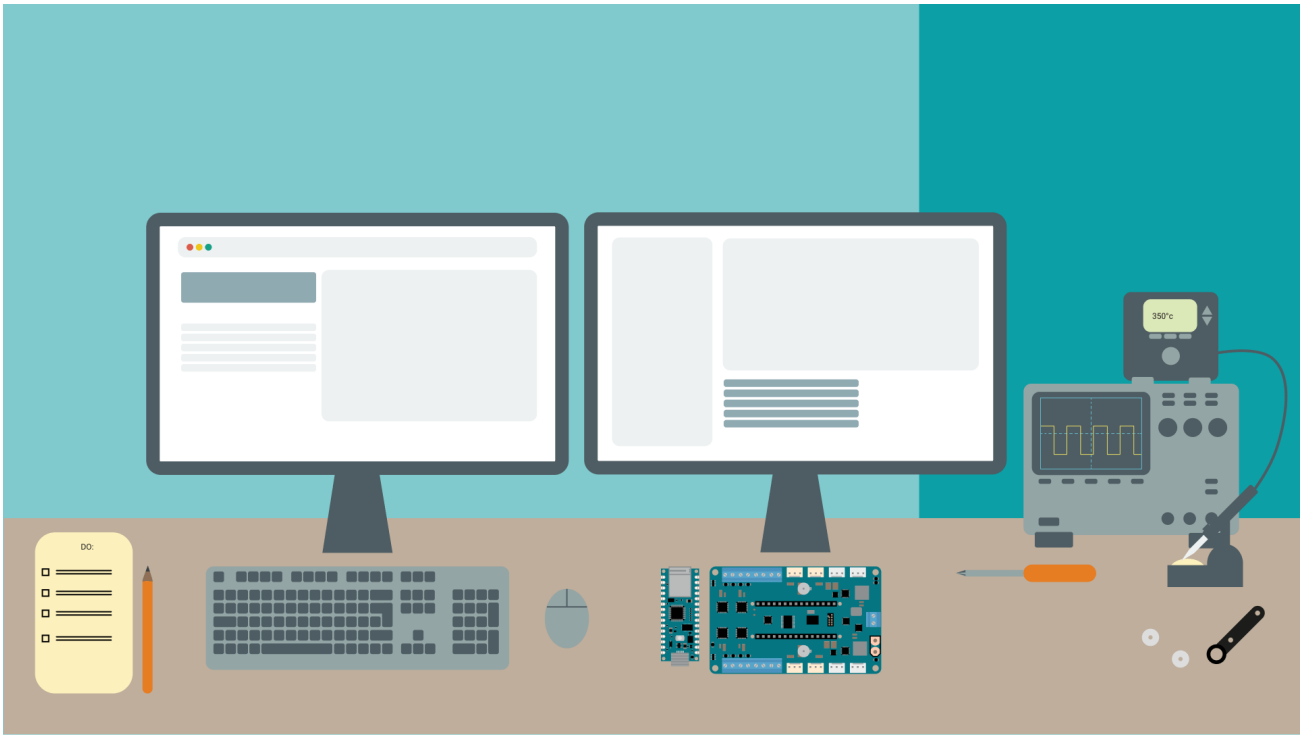
Simulink Support Package for Arduino extends Simulink with blocks for configuring Arduino sensors and reading and writing data from them. After creating your Simulink model, you can simulate it, interactively tune algorithm parameters until you get it just right and download the completed algorithm for standalone execution on your Arduino. With the MATLAB Function block, you can incorporate MATLAB code into your Simulink model.

Both Arduino support packages remove the need to manually write C or C++ code and compile, build, and run the application. The support packages translate your MATLAB and Simulink designs into working C and C++ code.

Using the kit

Before you start using the Kit and working with the projects, we have a few recommendations and guidelines that could make the learning experience better.

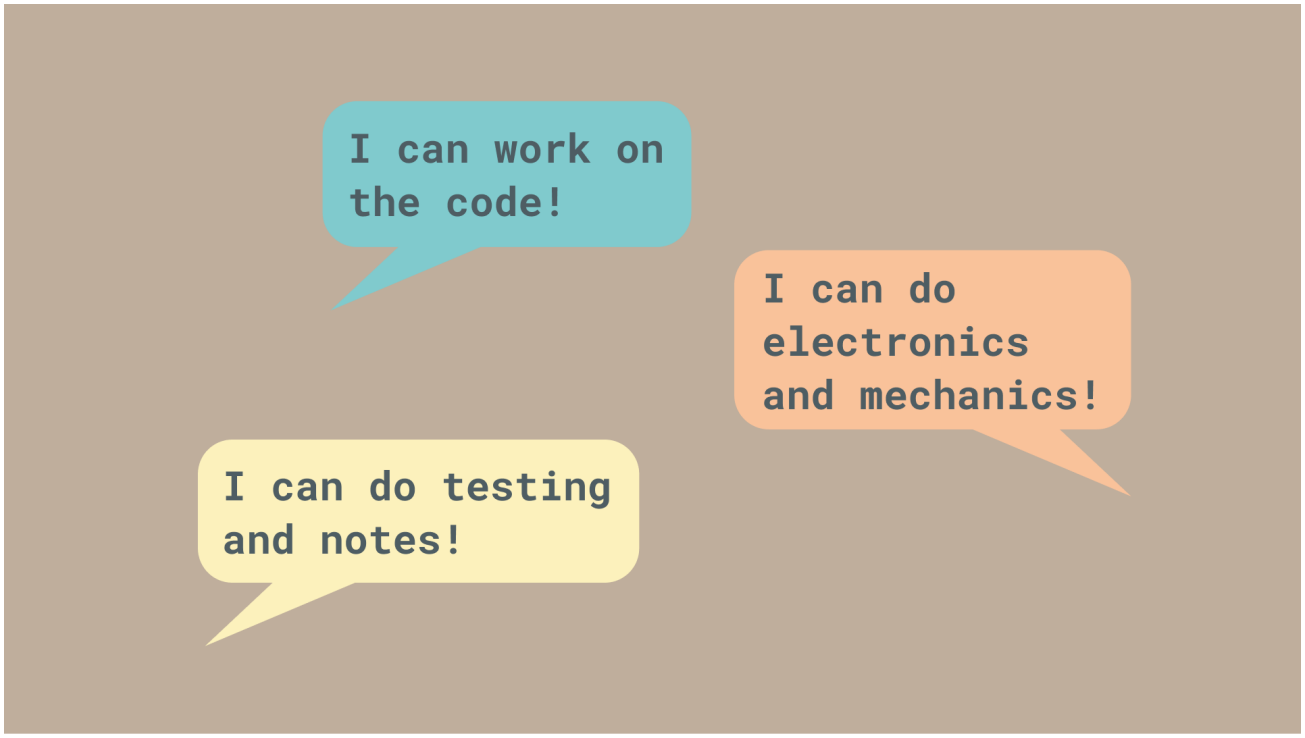
Setting up the Workspace



You will have to use both the online platform and MATLAB open at the same time, It is recommended to use two screens or two computers at the same time. Therefore we recommend setting an ample amount of deskspace when experimenting ar projects. You will need enough desk space to fit your computer, the hardwa

as have space for the final project to move around on the table. Make sure you also have a good Internet connectivity in your workspace. To access the online platform you will need to have an internet connection.

Assigning Roles of Teamwork

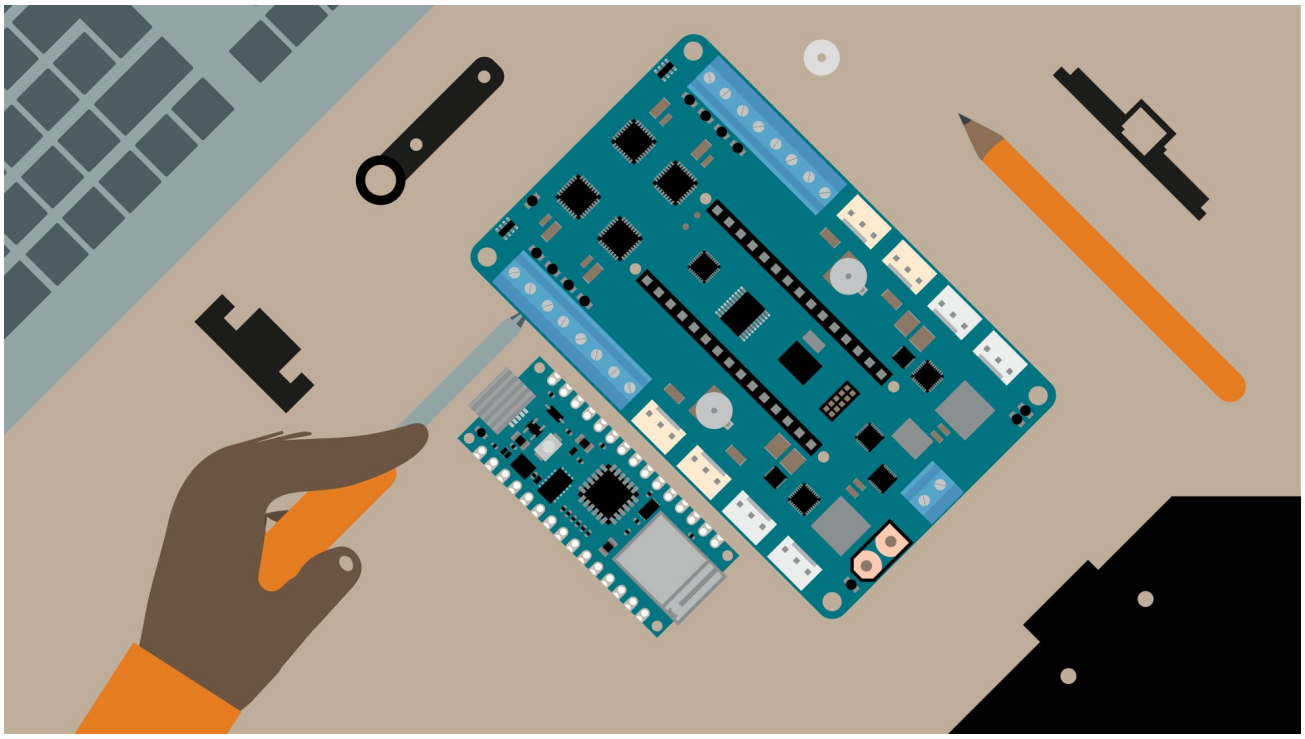


The kit is designed for two people (max three) before you start working as teams its important to assign roles. This is done to improve the team dynamics and collaboration while solving a problem together. Roles can range from developer, coder, problem solver and many more. The idea is to distribute responsibilities based on something that interests you and your teammates.

Planning and Organising your Work

It's a good practice to plan and organise your work before you start with each section. One way is to set realistic goals for yourself at the beginning of every session. To help you better plan we have defined the Learning objectives for each section in every chapter. It is also recommended to take a break as you proceed to keep the sessions more productive.

Keep Tinkering and Exploring



One of the characteristic aspects of the Self guided learning is experimenting while learning a topic. Try out the experiments to further your knowledge. You can also Use our extra learning materials in our References / resources section to further to take a deeper dive into certain areas of Mechatronics that aren't covered in this kit.

1.2 Learning Objectives

To make learning easier and graspable the kit is divided into 6 chapters where each chapter aims to achieve a particular set of learning outcomes. Learning objectives help you to realise your knowledge gaps and make an informed decision on what chapters you must take to fill this gap. It also allows you to conceive your learning goals and organise your sessions in a more efficient way. Every chapter in the Arduino Engineering Kit Rev2 is created to cover certain important aspects of skills and knowledge in relation to Arduino, MATLAB and Mechatronics.

Note: Each chapter is segmented in such a way that you have the freedom to choose how to progress through them.

Arduino, MATLAB & Simulink

In this chapter you will learn:

- How the entire Arduino platform works by learning the basics of the Arduino Nano 33 IoT Board, the Arduino Nano Motor Carrier and the Arduino IDE,
- About the MATLAB user interface, and how to create, access, modify and visualize data,
- The basics of MATLAB programming language and you will be able to write script and functions within MATLAB,
- How to create and simulate a Simulink model,
- How to visualize simulation data in the Simulink environment, set the sampling rate of a block in a Simulink model and add block hierarchy to a Simulink model using subsystems.

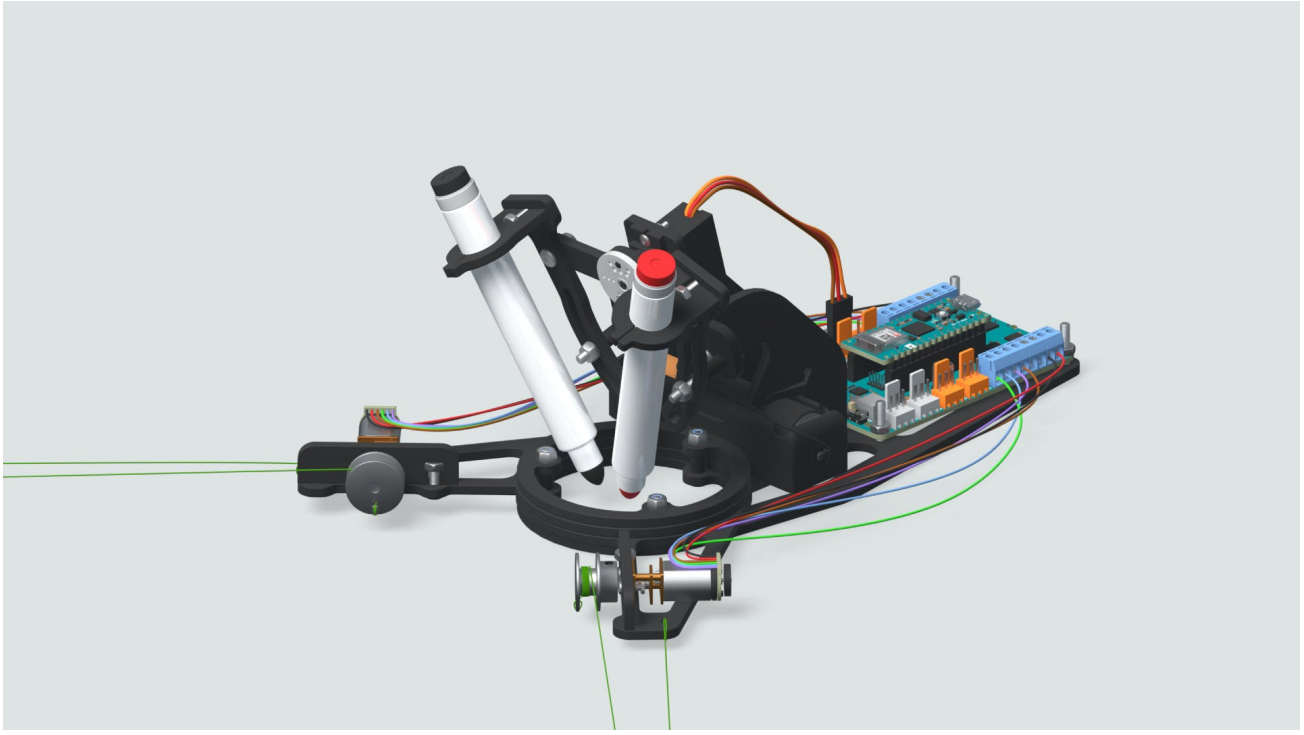
Basics of Mechatronics

This chapter covers the theoretical and practical concepts that will allow you to better understand the projects coming later. In this chapter you will learn:

- How DC motors, servo motors and encoders work,
- About electronic concepts such as I2C communication, PWM signals,
- How to control motors through MATLAB and Simulink,
- How to read values from the encoder and IMU sensors through MATLAB and Simulink.

Drawing Robot

Using image processing, trajectory calculation, and trajectory optimisation for path planning. The Drawing Robot draws an image on a whiteboard of an image of a picture from a webcam that is digitised by MATLAB. The robot utilizes pure MATLAB code and motor control system theory to convert a physical image to a digital format and onto a whiteboard.

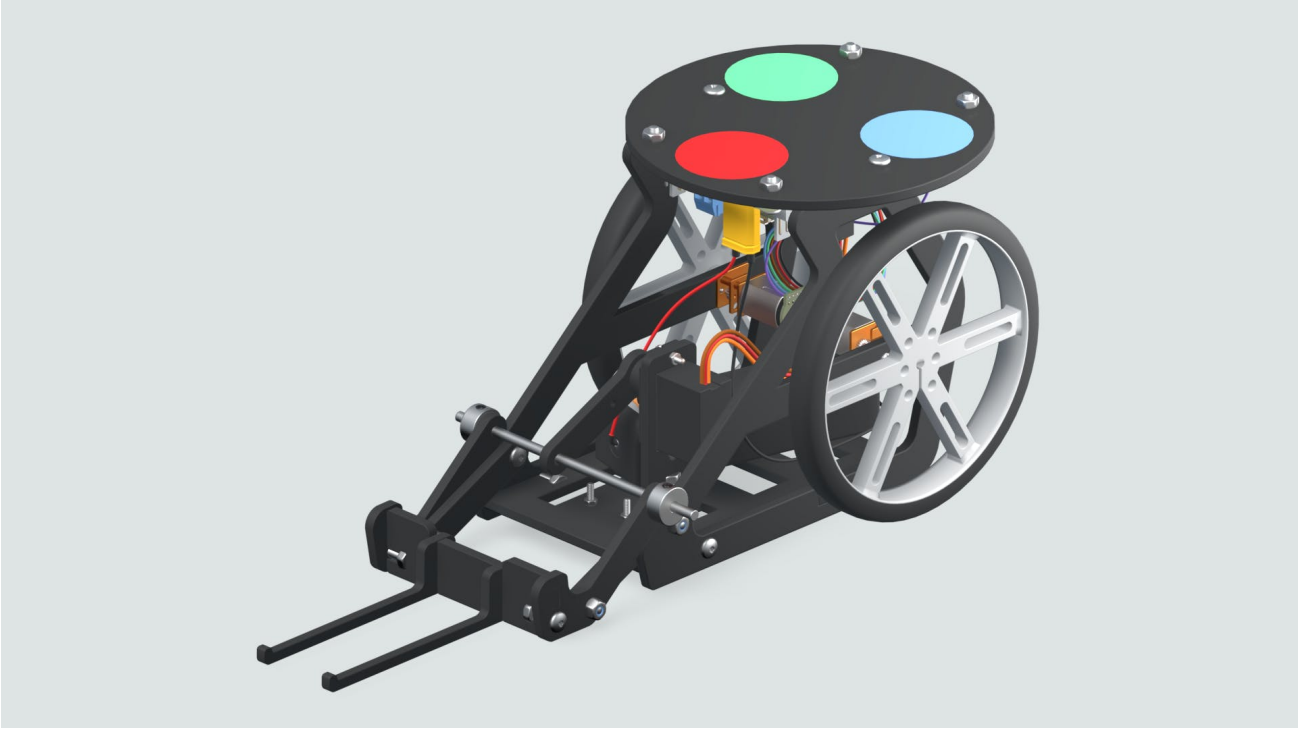


Through this project, you will learn

- How to connect to an Arduino-based robot from MATLAB,
- to write MATLAB apps, functions, and scripts to control your robot,
- to apply concepts from geometry, physics, symbolic math, and image processing,
- to automate a complete application workflow from start to finish.

Webcam Controlled Rover

The Webcam controlled Rover features position tracking via a remotely connected webcam and onboard sensing features for obstacle avoidance and movement. The rover uses a mix between MATLAB and Simulink programs to move around and interact with the world.



Through this project, you will learn:

- About the basic movement of differential drive robots from MATLAB,
- How to use kinematic equations to simulate the rover motion,
- Perform open and closed loop control of the rover,

- Use states to program your rover,
- Localization of the rover using image processing,
- Control the rover and forklift to pick up the target and drop it off,
- Wi-Fi communication between the rover and MATLAB.

Self Balancing Motorcycle

Featuring inverted pendulum dynamics using a reaction wheel and movement gyro, the Self-Balancing Motorcycle can move around and balance by itself. The motorcycle is using Simulink to monitor and control the movements and inertial sensing and filtering to make the motorcycle balance.

Through this project, you will learn

- How to simulate the vehicle's overall behavior and create models of the components to improve the quality of the simulation as well as the one of the control algorithms,
- How to program the motorcycle with Simulink®,

- How to control its balance algorithm and make it move in a straight line,
- How to build your own self-balancing vehicles.

1.3 Installing the Tools

The Engineering Kit Rev 2 uses MATLAB version **2021b** throughout the kit. In this section you will install the MATLAB, Simulink and other necessary Add-Ons. The installation process is divided into 5 stages, where the first one is the installation of the MATLAB & Simulink tools followed by the steps to install the Add-Ons.

Note: If you already have a registered MATLAB and Simulink (R2021a and below), please make sure you to update the software to R2021b. If you already have the R2021b you can jump directly to step 2.

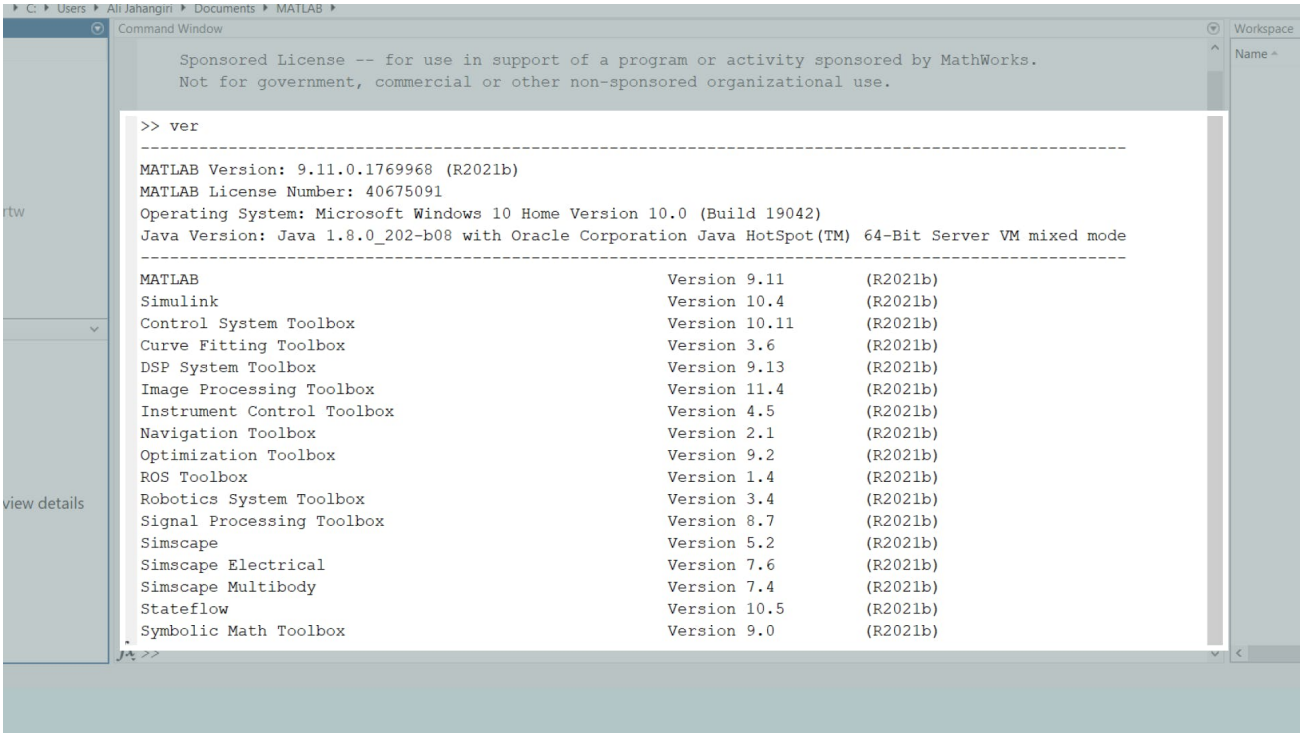
You will need the following MathWorks products to program the projects in the Arduino Engineering Kit Rev2:

- MATLAB®
- Simulink®

- Control System Toolbox™
- Curve Fitting Toolbox™
- DSP System Toolbox™
- Image Processing Toolbox™
- Instrument Control Toolbox™
- Optimization Toolbox™
- Signal Processing Toolbox™
- Simscape™
- Simscape Multibody™
- Stateflow®
- Symbolic Math Toolbox™

Self-balancing Motorcycle	Webcam controlled Rover	Drawing Robot
MATLAB	MATLAB	MATLAB
Simulink	Simulink	Control System Toolbox
Simscape	Stateflow	Image processing Toolbox
Simscape Multibody	Image processing Toolbox	MATLAB Support Package for USB Webcams
Simulink Support Package for Arduino Hardware	Instrument Control Toolbox	MATLAB Support Package for Arduino Hardware
	MATLAB Support Package for Arduino Hardware	
	Simulink Support Package for Arduino Hardware	

The kit includes a one-year individual user MATLAB license that includes all of these products. If you have an existing MATLAB license, you can use that as long as you have the required products and access to release R2020b or later. You can see if you have the required products installed by typing `ver` in the **Command Window**.



If your school has an active Campus-Wide License, it should have the required products and access to R2021b. You can check if your school has a campus license **here**.

If you do not have an existing license and the included license is not sufficient for your intended use (e.g. single kit will be shared by multiple students and/or the kit will be used for longer than one year), **contact MathWorks**.

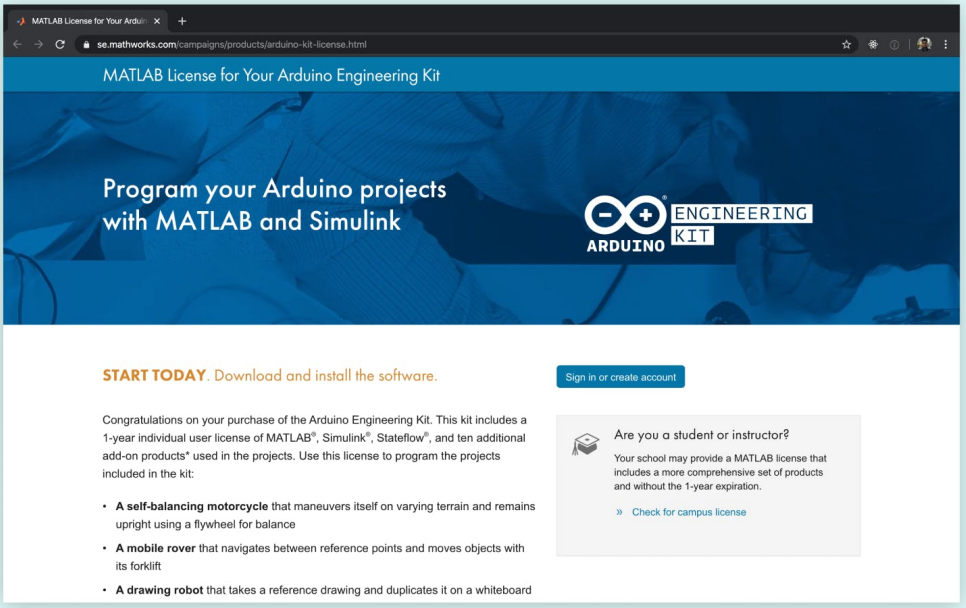
Installing MATLAB & Simulink

The following steps guide you through the installation of the MATLAB license included in the kit.

> **Note:** If you've already installed MATLAB with a valid license, you can skip this section. Sign into your MathWorks Account and jump to the next section, "Installing MATLAB support package for Arduino hardware". You can visit this **page** to verify what license you have for MATLAB.

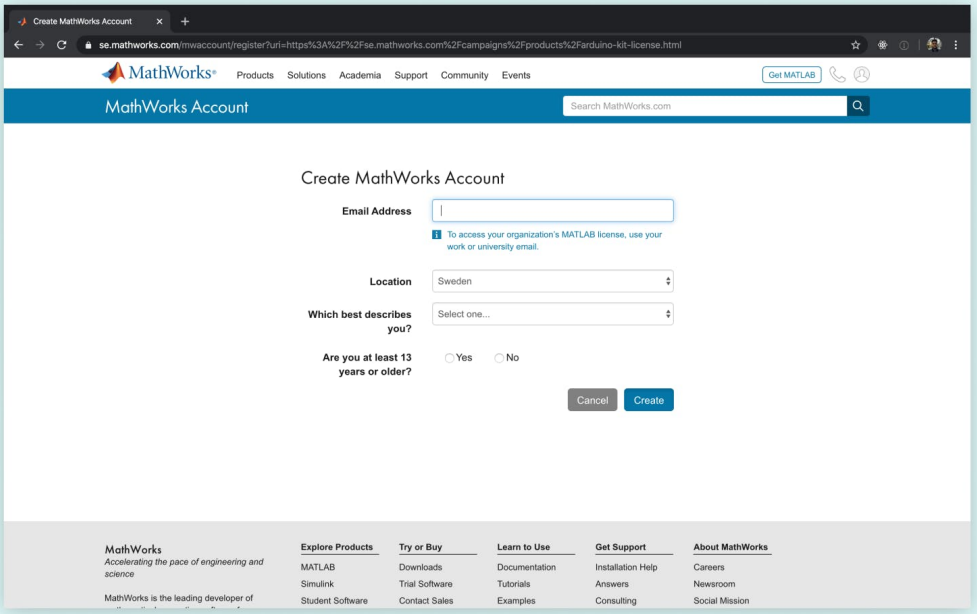
1. Visit MathWorks Website Help

Visit **mathworks.com/arduino-kit-rev2-license** and you will be prompted to create a MathWorks Account (or sign into your existing account), agree to Terms of Use, and enter the code from your kit.



2. Create / Sign in to your account

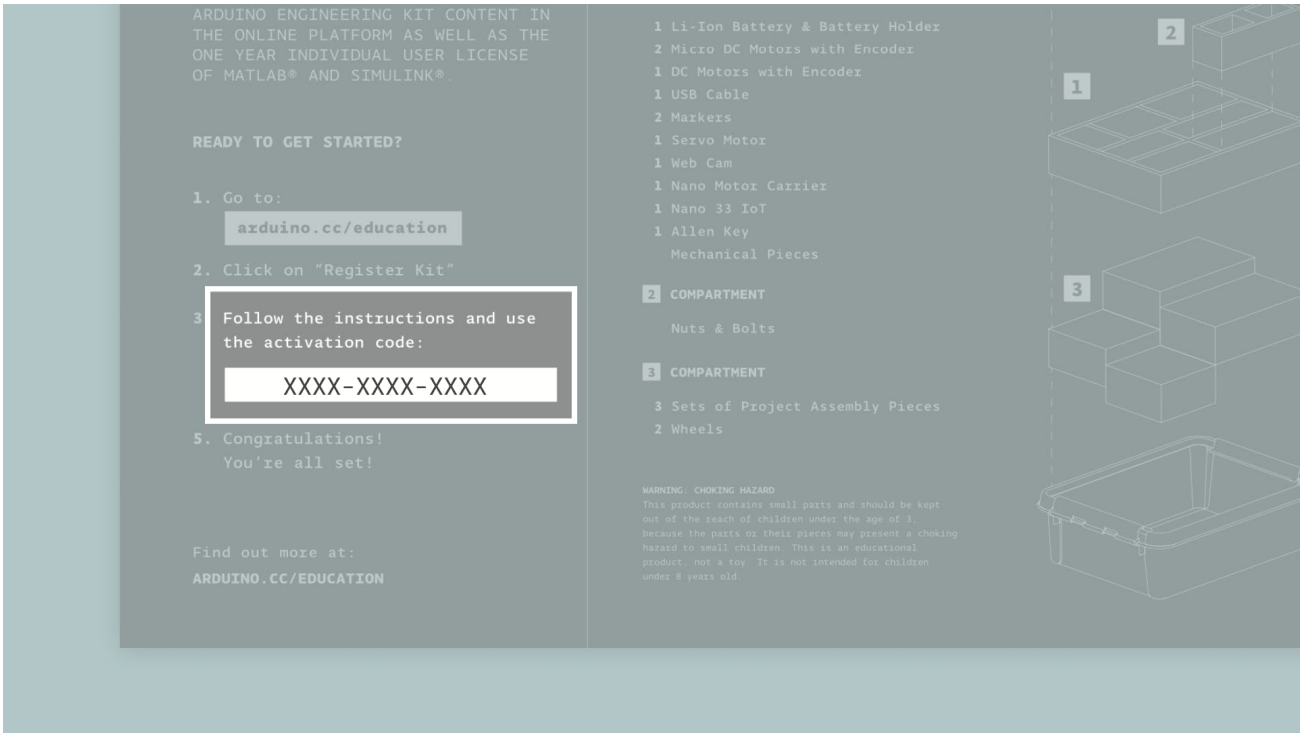
Enter the details to which you want the license to be connected to. If you already have an account, sign into your account.



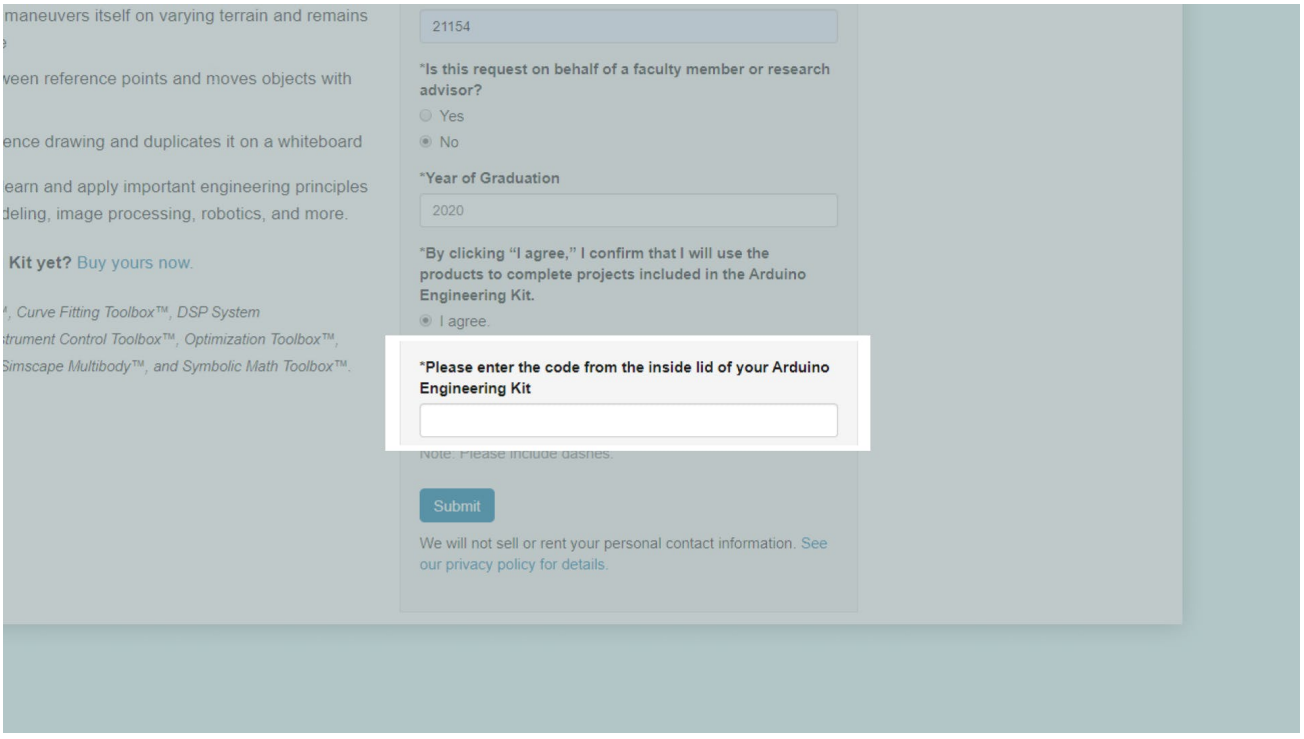
Note: For those using the kit as a group, we would recommend that you use an email ID that's accessible to all the members in the group.

3. Enter your license

The Arduino Engineering Kit Rev2 comes with one individual license to MATLAB. The activation code can be found on the information sheet inside the box.



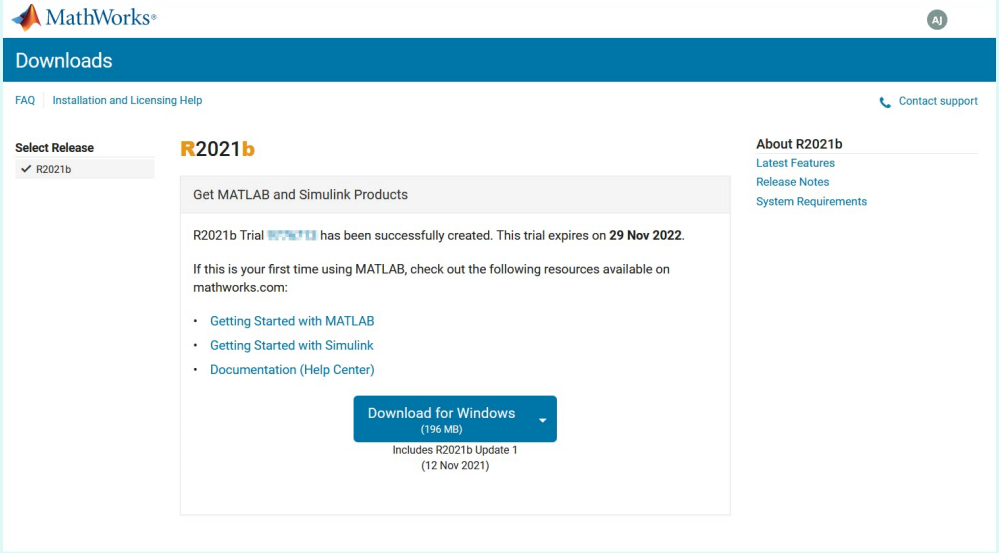
Enter the activation code from your kit in the space provided on the registration page and click **Submit**. After submitting the required information, your license is immediately created and available for installation.



Note: The kit comes with one individual MATLAB license, if you want to have licenses for multiple users you find more information about different license types and pricing from **here**.

4. Download MATLAB

If you are ready to install now, choose your platform and proceed with the installation. Follow the instructions **here** to install.



If you decide to install later, visit **mathworks.com/mwaccount** and choose the license number that corresponds to the MathWorks Arduino Kit.

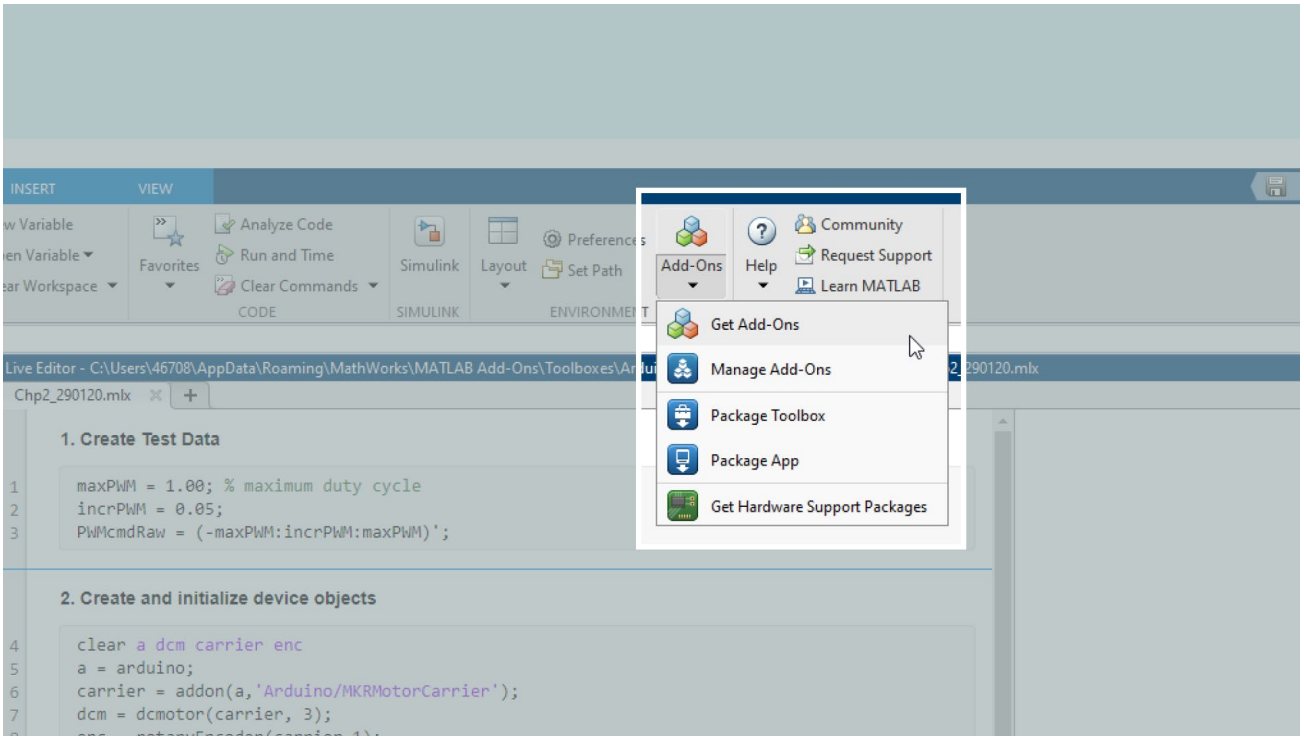
Installing MATLAB Support Package for Arduino Hardware

You will need to install the support package for the Arduino Hardware. The following steps guide you through the installation for which you will also need a Arduino Nano 33 IoT board and an USB cable.

Note: The MATLAB Support Package for Arduino Hardware installs Arduino IDE 1.8.10 and a few associated libraries. Do not alter these files or their location.

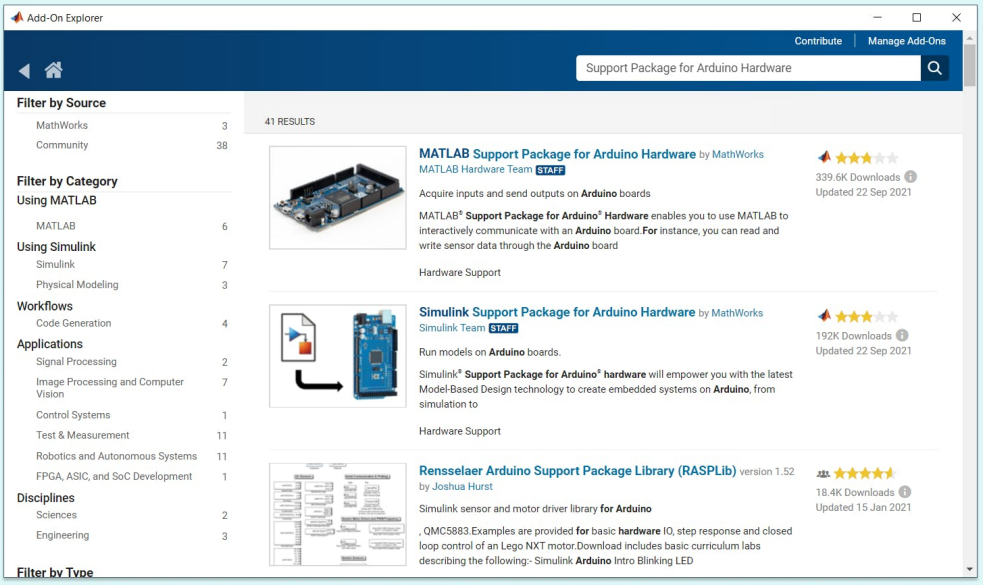
1. Open Add-Ons

The **MATLAB Support Package for Arduino Hardware**: Acquire inputs and send outputs to Arduino boards and connected devices. Open MATLAB with Administrator access. Select **Add-Ons** from the **MATLAB Toolstrip** as shown in the image below and click on Get Add-Ons.



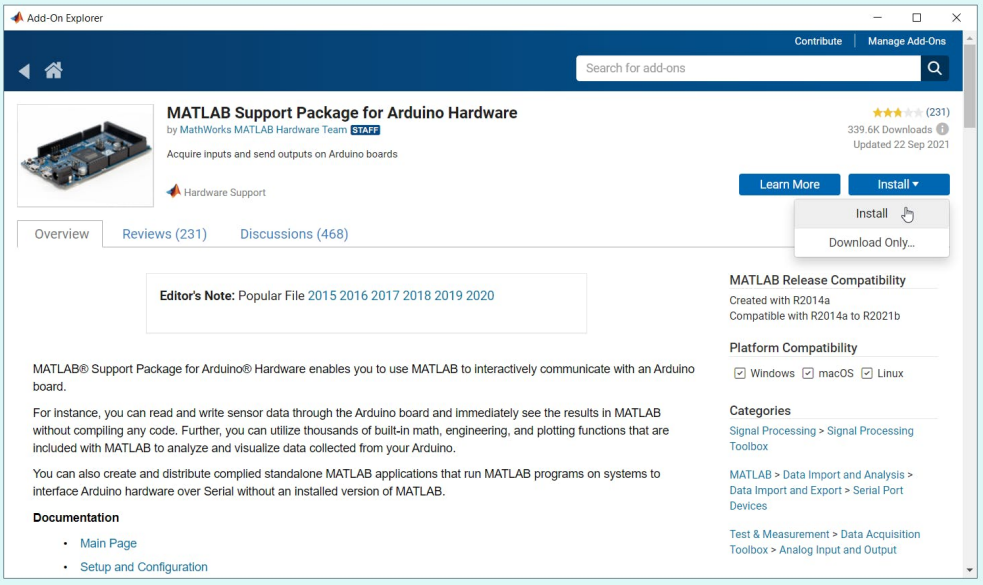
2. Search for the Add-on

In the **Add-On Explorer** window, search for **Support Package for Arduino Hardware**. Select **MATLAB Support Package for Arduino Hardware**.

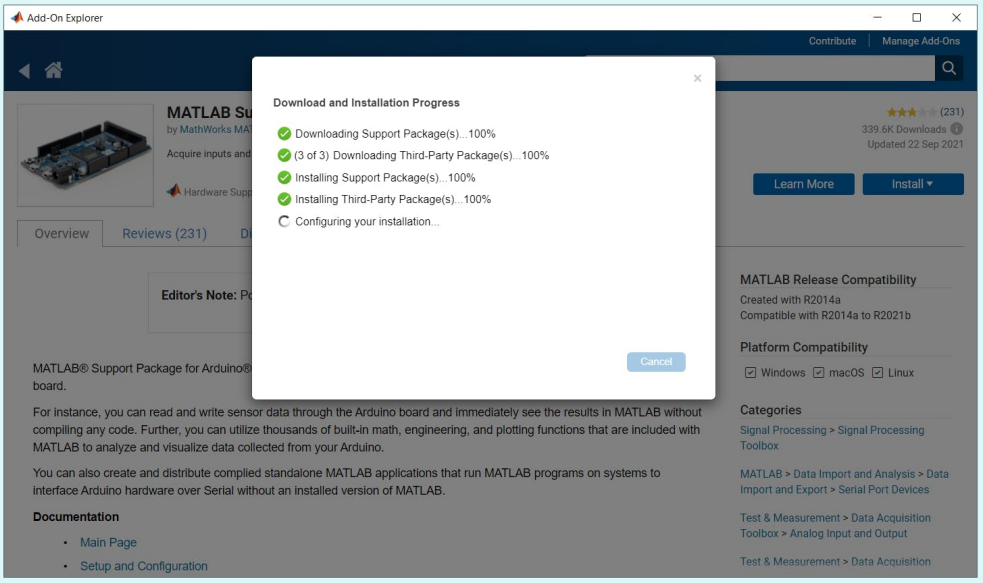


3. Install the Add-On

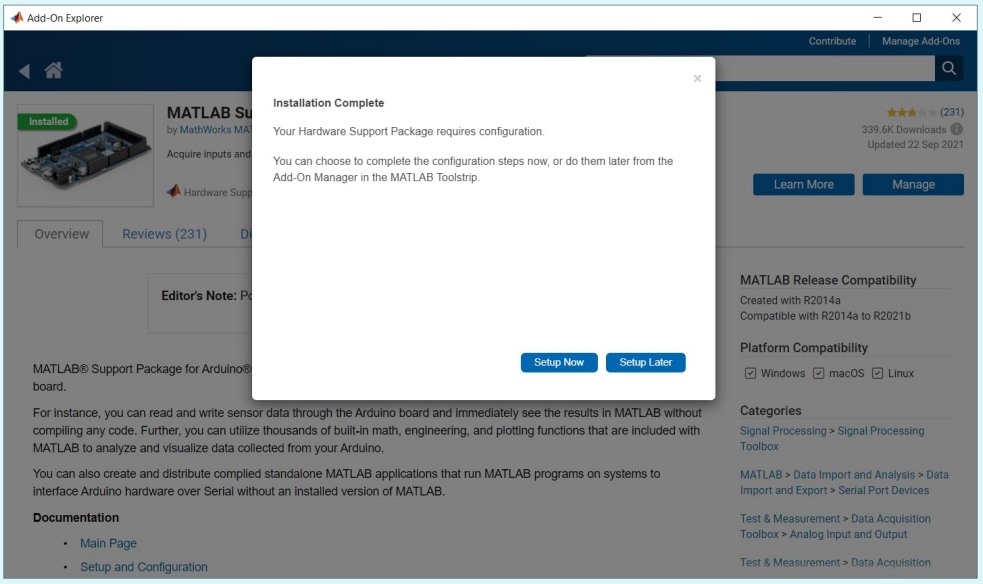
Click on **Install** then in the drop down menu that appears select **Install** again.



You may be asked to sign in. The installation process will take some time depending on the speed of your internet connection. Make sure you have a good internet connection at this stage.

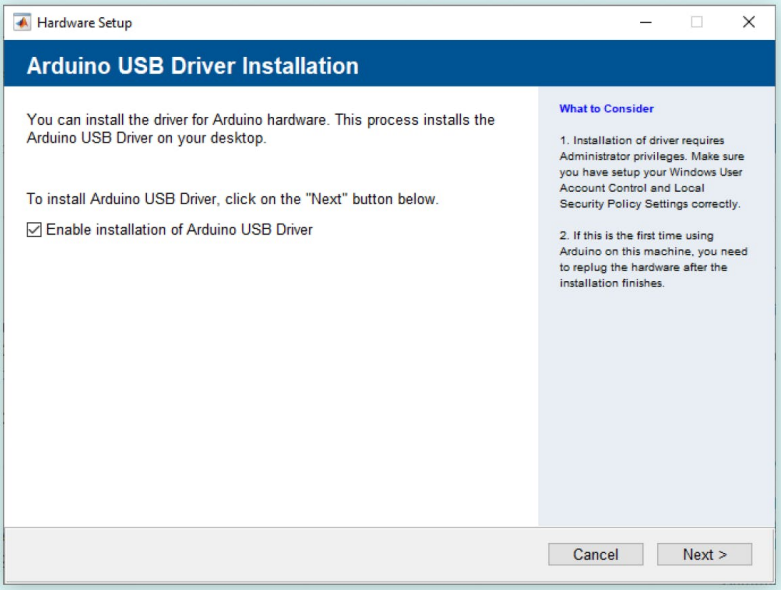


Once the installation is complete, click on **Setup Now** to proceed with the next step.



4. Arduino USB Driver Installation

Since the Arduino Hardware connects to the MATLAB tool through USB, you must install the Arduino USB Driver.

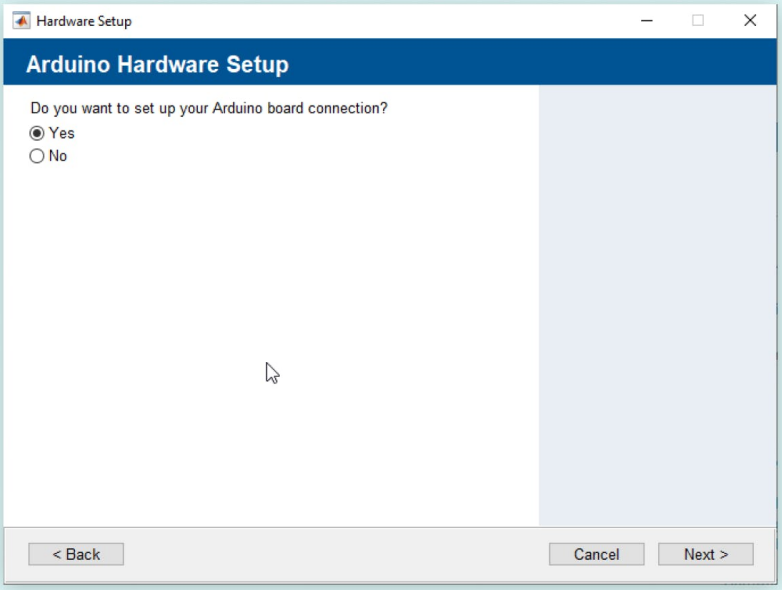


Testing the MATLAB-Arduino USB Connection

MATLAB is now ready to interface with the Arduino ecosystem, including the Nano 33 IoT. It is useful to test the USB connection to make sure everything runs smoothly.

1. Enable the USB Driver

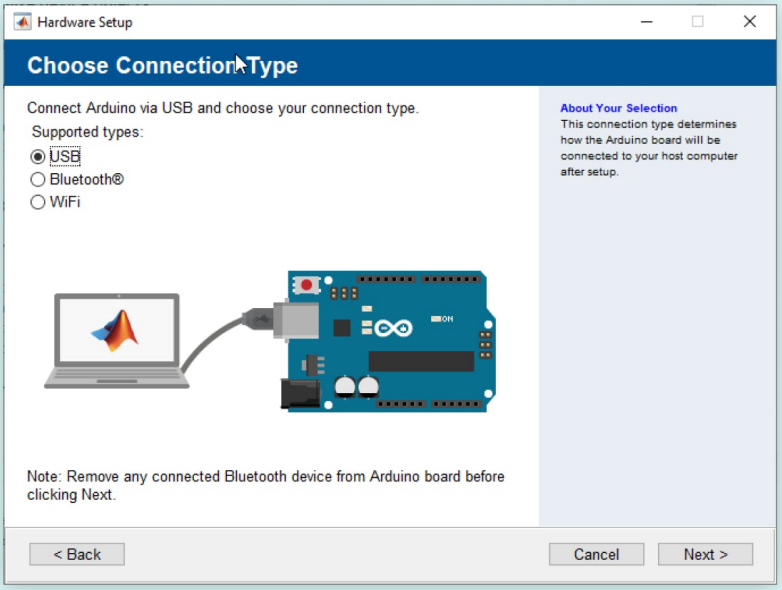
In the **Arduino Hardware Setup** screen, make sure that the **Yes** radio box is selected and click on **Next**.



If you have closed the window after installing the USB driver, you can enter `arduinosetup()` to bring up the GUI for the Arduino Hardware Setup.

2. Connect the Board

Now connect your Arduino Nano 33 IoT board to your computer using a USB cable and choose **USB** as the connection type. Click on **Next**.

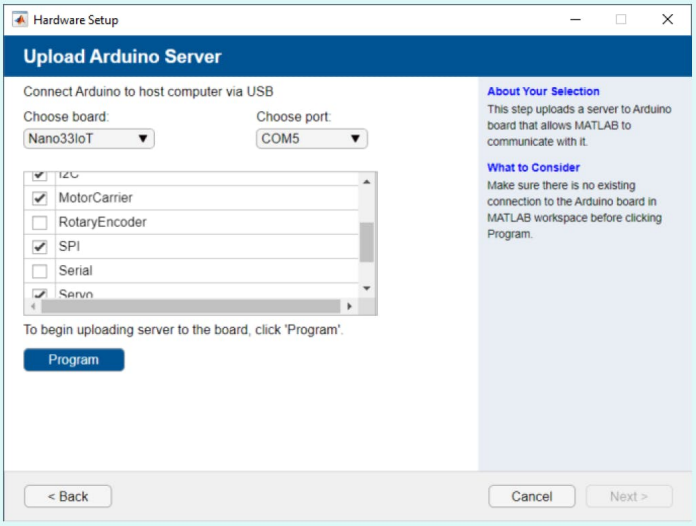


3. Select the Board

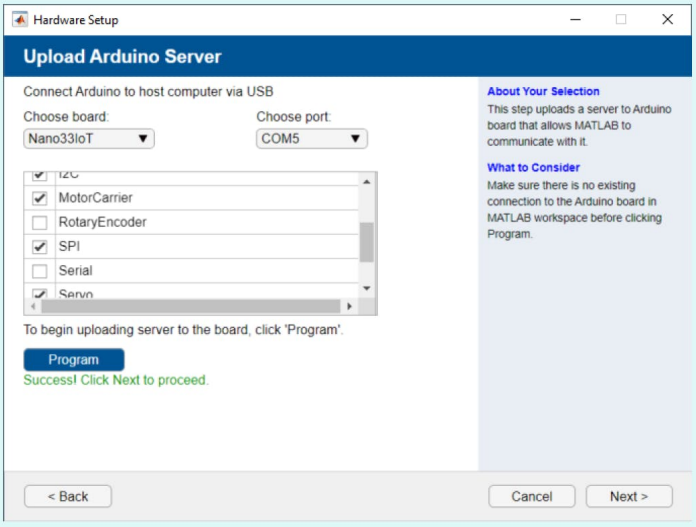
Choose the board as **Nano33IoT** and choose the corresponding port number to which the board is connected to. The following libraries are used in the projects:

- I2C
- MotorCarrier
- SPI
- Servo

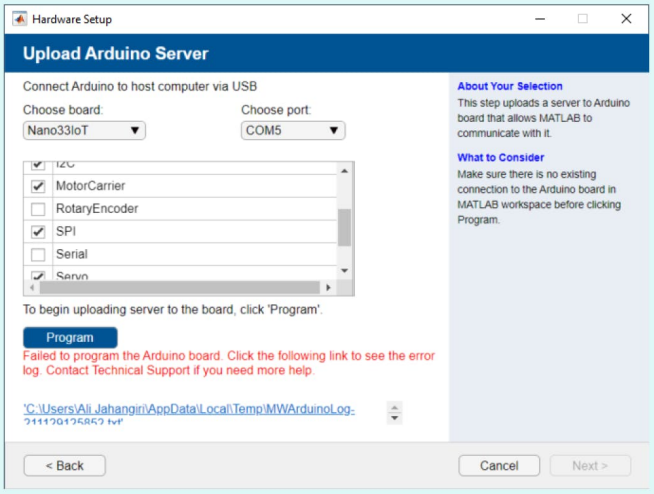
Once you have selected the libraries, click **Program** to upload the Arduino Server to the board.



It may take a few minutes to compile and upload the Arduino Server code. When it is uploaded, you will see a message confirming this under the program button.

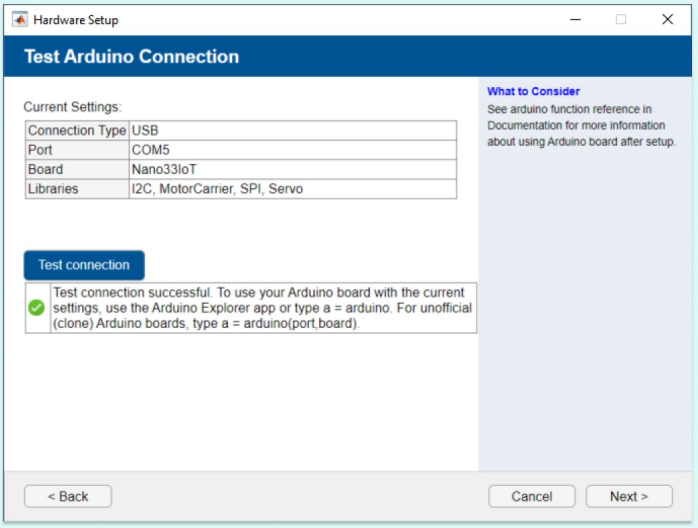


Note: Should the connection fail, you will see a message stating this. A .txt log file is generated. When contacting Support, please share this file so that we can help you faster.



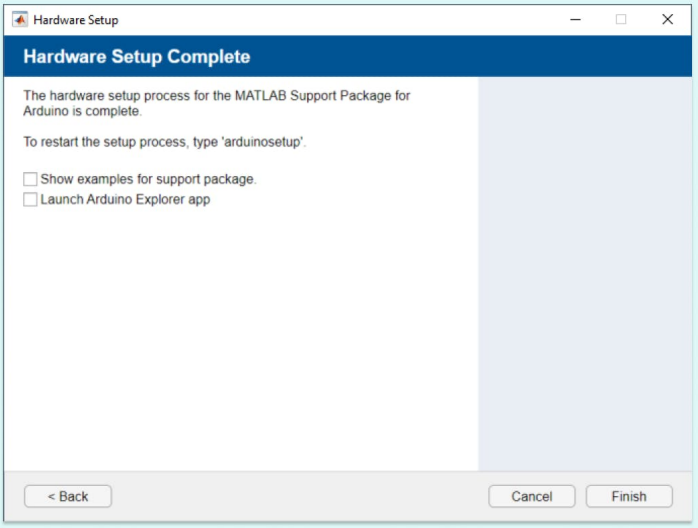
4. Test the Connection

It is advised to test the connection between MATLAB and your Arduino Nano 33 IoT before you proceed with the projects. Click on **Test Connection** and verify that the connection is successful.



9. Complete the Setup

When the hardware setup is complete you will get a notification marking the completion of the process. You can choose to download the examples using this support package as a reference for your other projects or view the Arduino Explorer, but this isn't mandatory.

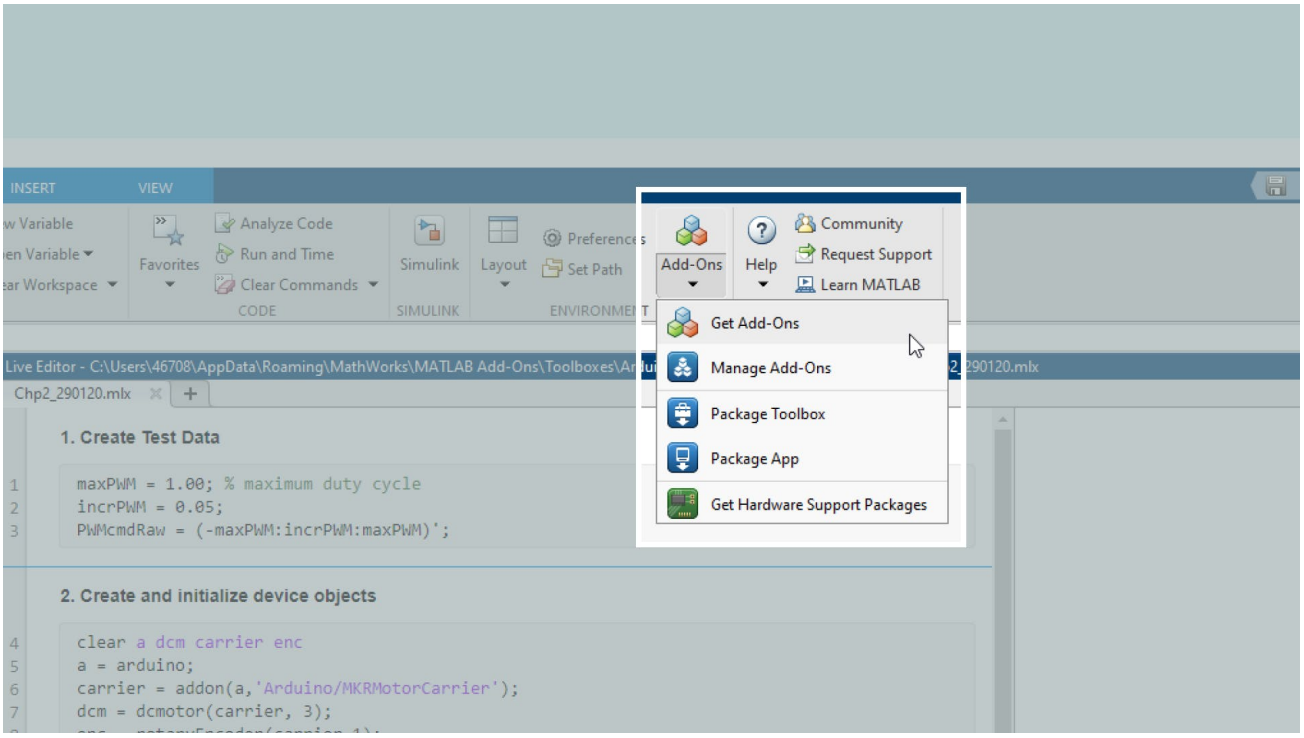


Installing Simlink Support Package for Arduino Hardware

The **Simulink Support Package for Arduino Hardware** is used to run the Simulink models on Arduino boards.

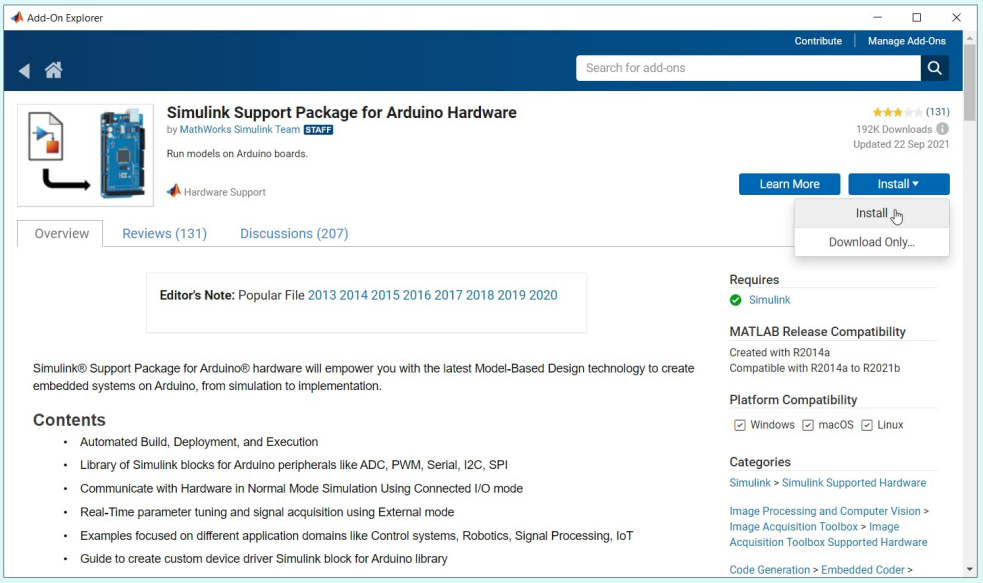
1. Open Add-Ons

Open MATLAB with Administrator access. Select **Add-Ons** from the **MATLAB Toolstrip** as shown in the image below and click on **Get Add-Ons**.



2. Search for the Add-On

In the **Add-Ons** window, search for **Simulink Package for Arduino Hardware** and click on **Install** then **Install** again.

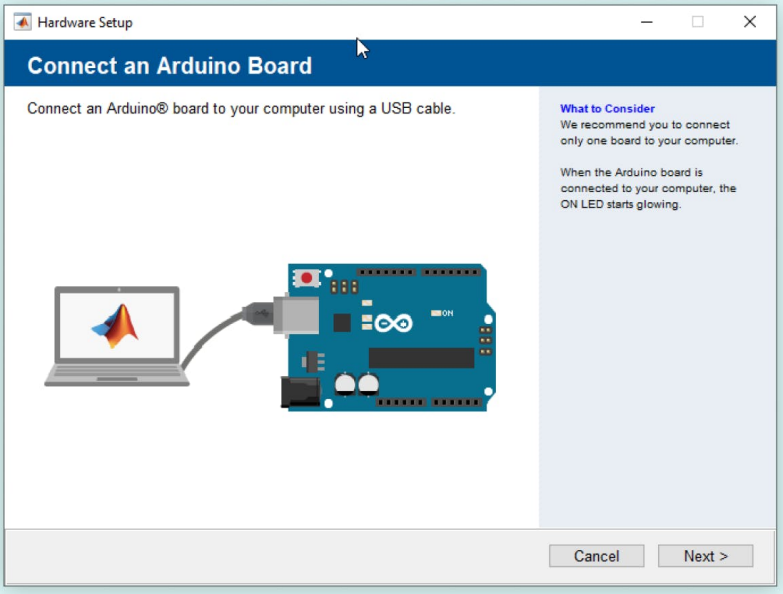


The installation process will take some time depending on the speed of your internet connection. Make sure you have a good internet connection at this stage. Once the installation is complete, click on **Setup Now** to proceed.

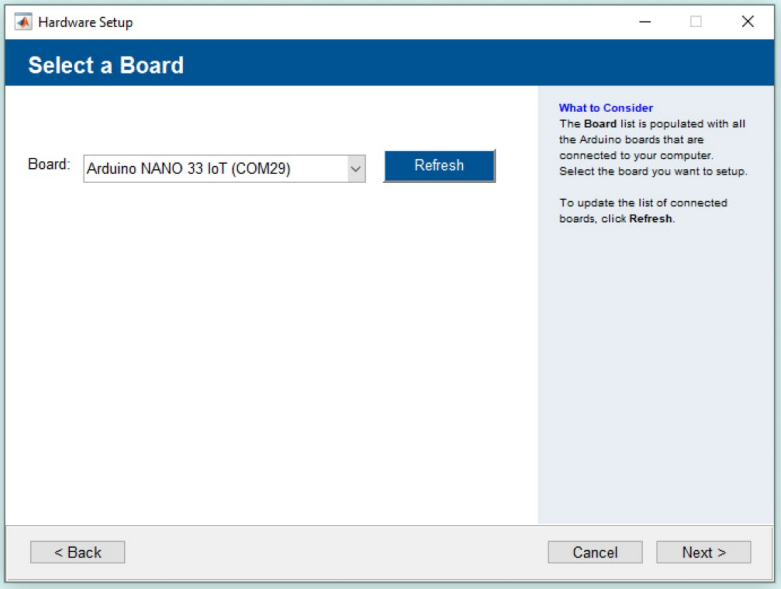
Testing the Simulink-Arduino USB Connection

1. Start the Hardware Setup

Additional third-party tools may need to be installed before you can continue. We will now test the Simulink-Arduino connectivity. Make sure the Nano 33 IoT is connected and click on **Next**.

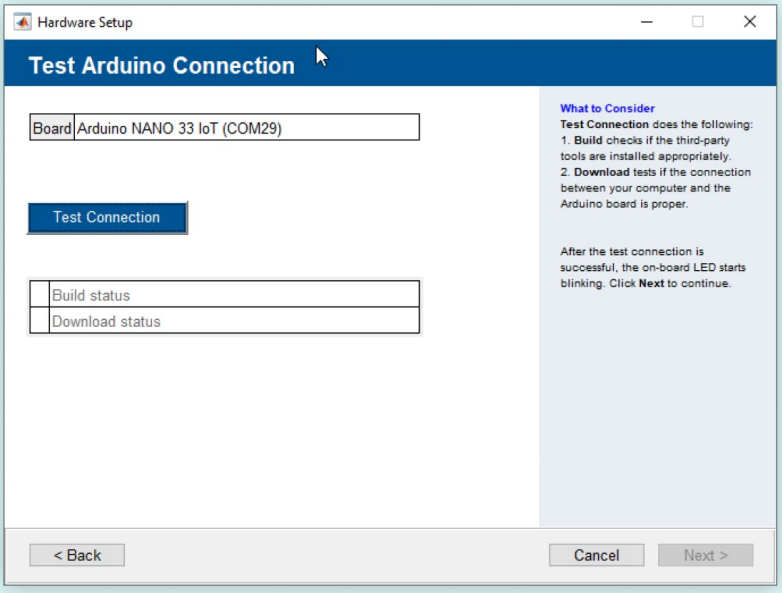


Next select Arduino Nano 33 IoT board from the options and click **Next**. If you can't find the name, click **Refresh**.

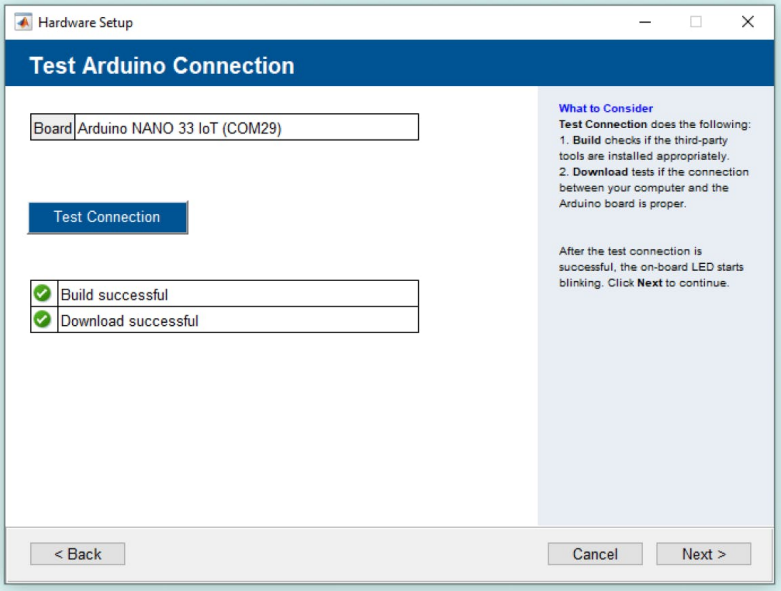


2. Test the Connection

When asked to verify the setup, click on **Next** to proceed. This is done to check if the third party tools are installed properly and if the connection between the Arduino board and the computer is proper. Click on **Test Connection**.



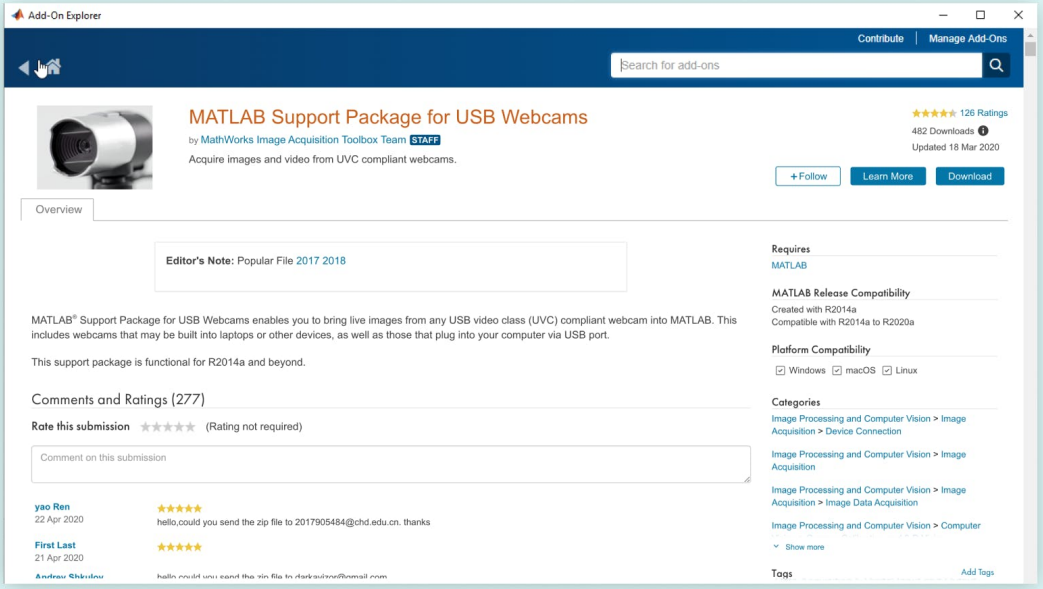
Once the test is successfully completed, the board's built in LED starts blinking. At this point you can click next to end the setup process.



Installing Support Packages for USB Webcams

The **MATLAB Support Package for USB Webcams** is used for acquiring images and video from USB webcams.

From the **Add-ons** window, search for the **MATLAB Support Package for USB Webcams** and install it. This Add-On is smaller in size and has a relatively shorter installation time.

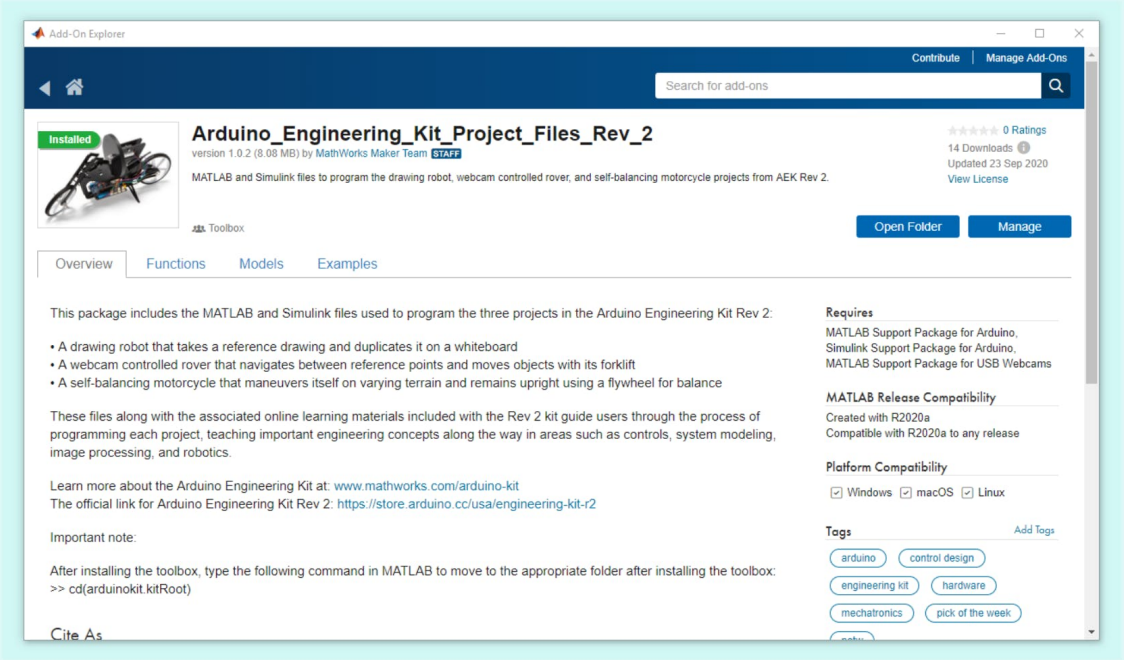


Installing Project Files

The **Arduino Engineering Kit Project Files** are MATLAB and Simulink files to program the three projects in the Arduino Engineering Kit Rev2: Self-balancing Motorcycle, Webcam controlled Rover, and Drawing Robot.

1. Search for the Add-On

In the **Add-Ons** window, search for **Arduino Engineering Kit Rev2 Project Files** and click on **Add**.



Note: Starting in MATLAB R2021a, the project files have been integrated within the MATLAB & Simulink Support Package for Arduino. For more details please visit this **link**.

Now you have installed MATLAB, Simulink, all the required Add-Ons and the Arduino IDE. In the next chapter you will start using these tools.