

# Teacher Info

TRY IT FOR:



AI Sign Train is an activity that combines learning about AI with coding the intelino train using Scratch. Students will control the intelino train with signs that they train the computer to recognize. This activity can be used for the Hour of Code™, since it is as interesting for beginning coders as for advanced ones. However, neither kids nor teachers need to know anything about AI, Scratch, or the intelino smart train!

The activity is designed to be self-guided and works for grades 2-8, although younger kids may need some help. Students will learn how to use the AI Scratch extension "Teachable Machine", developed by Google. They will create and train a model to make the computer recognize and differentiate signs. Then, students will code in intelino Scratch to control the train.

For intelino Scratch, please check [here](#) to learn more about intelino Scratch and its requirements. The Teachable Machine extension is directly available in intelino Scratch, but note that it is currently not supported on iOS.

Note that students start coding with a sample program that can be downloaded to their device and then opened in Scratch, or simply assembled from the blocks during the activity. (download the program [here](#))

If you are new to teaching with intelino, take a look at our [Teacher's Quick Start Guide](#). We also have more lessons and activities (unplugged or using Scratch or Python) in the [intelino lab!](#)

## Prerequisites

none - no knowledge of coding or intelino is required

## Code Modes

mode 3 (intelino Scratch)

## Group

up to 4 students per group

## Time

1 session

## Grades

2-8

## Supplies

per group:

- 1 intelino engine, charged
- device
- printout of pages 1-12
- printout of the stations
- printout of the signs, must be in color
- scissor

## Standards

- CSTA: 1A-AP-8, 1A-AP-10, 1A-AP-14, 1B-AP-10, 1B-AP-12, 1B-AP-15, 1B-IC-18
- Common Core: CCSS.MATH.PRACTICE.MP1, CCSS.MATH.PRACTICE.MP3
- ISTE: 1.1.c, 1.7.b, 1.7.c

Questions?

email [julia@intelino.com](mailto:julia@intelino.com)



**AI  
SIGN  
TRAIN**  
An AI activity

**Ages:**  
8+  
**Difficulty:**  
beginner  
**Code Modes:**  
mode 3 - intelino Scratch  
🕒 1 session

**TRY IT FOR:**  
**HOUR  
OF  
CODE**

**intelino**<sup>®</sup>

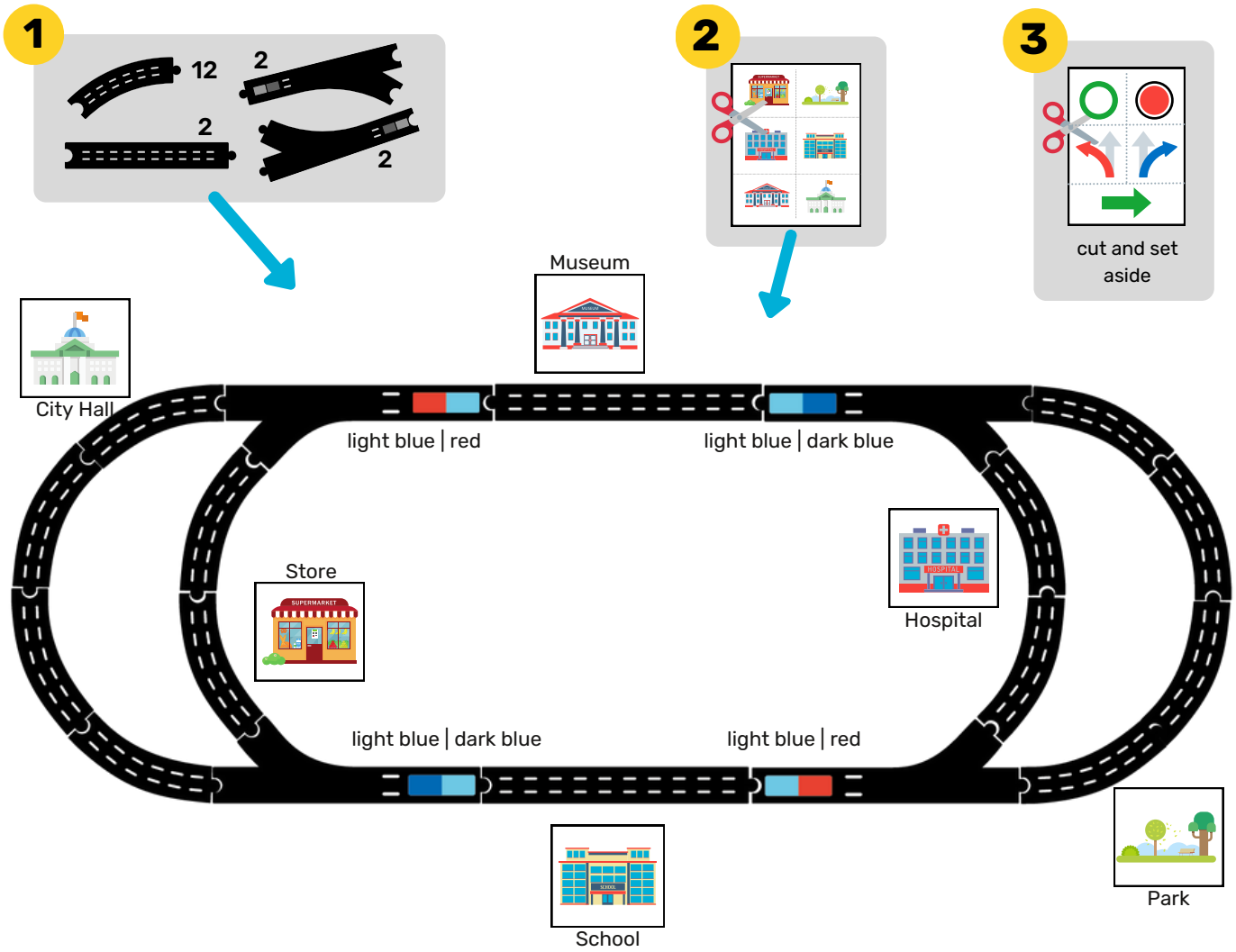
NAME \_\_\_\_\_



**Hi, I'm Jamy! I'm an intelino smart engine.**

I'm an autonomous train, which means I don't have an engineer that drives me. You will be controlling me today.

How are you going to do that? With signs, just like a real-life conductor!



**GOAL**

Make Jamy drive and stop at every station!

**HOW?**

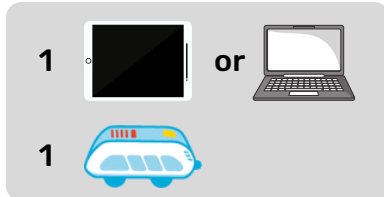
Make a program that recognizes signs and makes Jamy drive and stop depending on the sign. Then control Jamy to stop at every station.



## Before we get to program, we have to prepare.

Let's start intelino Scratch and connect it to the engine. We will also add another extension called Teachable Machine.

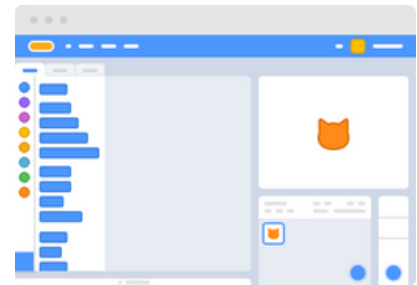
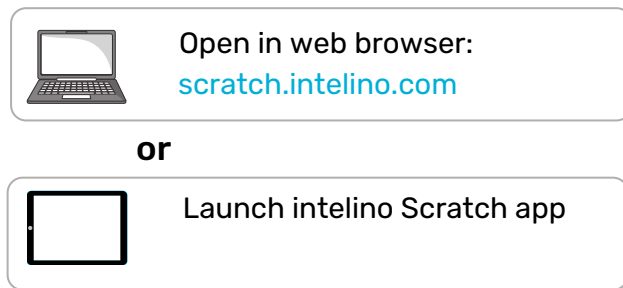
### 1 Get engine and device



### 2 Power on engine

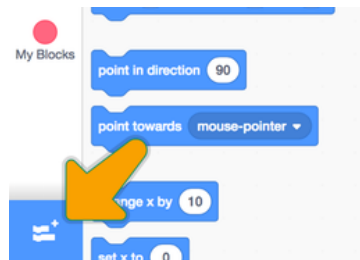


### 3 Start intelino Scratch



### 4 Open Scratch Extensions

Click on the blue button in the lower left corner

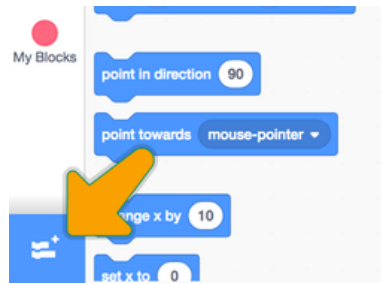


### 5 Select Teachable Machine



## 6 Open Scratch Extensions again

We have to add another one!

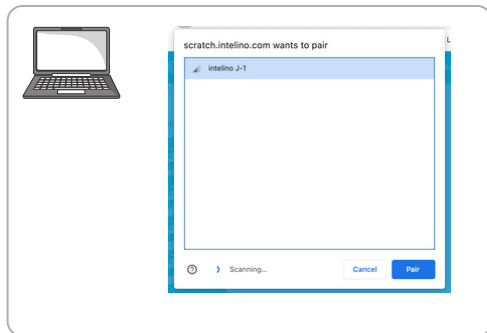


## 7 Select intelino smart train 1

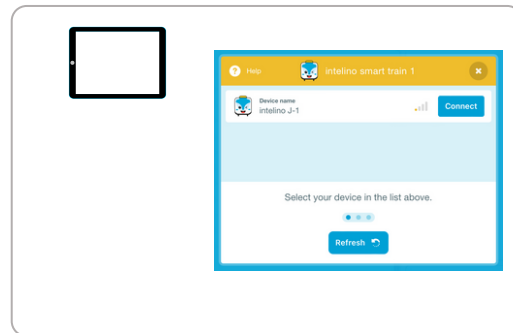


## 8 Connect engine

Find your engine and **Pair** or **Connect**

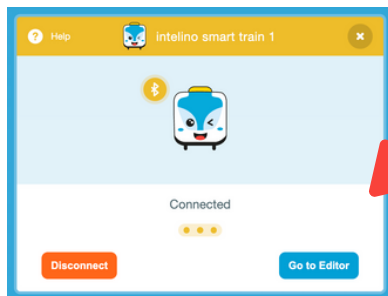


or

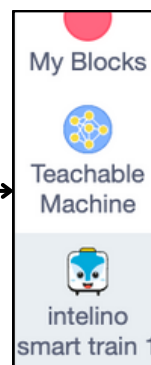


## 9 Success!

Now go to the Editor



editor



You can see the two new extensions in the left side bar



## Now let's look at our task:

How can we control the train with our signs?



### Teachable Machine

It's a program that you can use to train your computer to recognize your own images, sounds, and poses.

When certain images (signs) are detected, then control the train with scratch blocks



### intelino Scratch blocks

Commands that we can use to make control the train (make it do certain actions).

### Example

When Teachable Machine detects a green circle



then make the train drive.



Code Talk:  
The **When** is call an event, and  
the **Then** is the response!



## But how do we train the Teachable Machine to recognize our signs?

Let's take a look!

### How does Teachable Machine work?

Teachable Machine is an **Artificial Intelligence** program, or **AI** for short. It will be our assistant in recognizing our sign images.

### But aren't computers bad at things like recognizing images?

True, computers are bad at some things that humans can do much better, but we will teach it to recognize our signs. This is called **creating a model**.

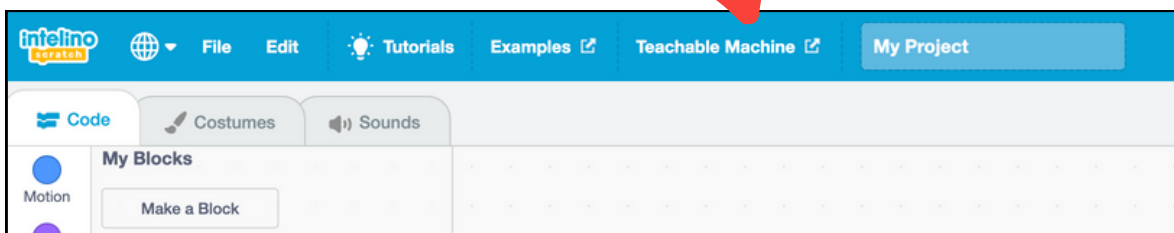
We will do these three steps:

1. Give examples of different signs to the model.
2. **Train the model** and check if it's doing what we want it to.
3. Export the model so we can use it to code in intelino Scratch.

## Let's create a model!

▶ **Go to Teachable Machine**

Click here



▶ **Start a new project**

Select "Image Project", then "standard image model".

### **Make the first class**

Classes are how we define different things that we want to have the computer recognize. Let's start with starting and stopping to drive. We want to make one class called "Drive" to recognize the drive signal. That's the one with the green circle. Follow these steps:

- Edit "Class 1" to be called "Drive".
- Click on the webcam sign.
- Take the green circle sign and hold it up to the camera. Now click on "Hold to Record" and keep on holding it. Move the sign towards and away from the camera. This way, the model can see how it looks like. You want to give the model good samples of how it looks like when you are holding the sign up to the camera so it can more reliably tell when it's happening.
- You can repeat recording until you are happy with the sample images. And don't worry - you can always add more samples later to refine your model.

### **Make another class**

Make a second class called "Stop" for the red circle stop sign.

### **Train your model**

Click on "Train Model". Now the tool is working. The Teachable Machine is learning from the samples and is creating the model. Do not move away from the tab while this is going on!

### **Tweak your model**

Go to the Preview window on the right. Try holding up the signs and see what the model predicts. At this point, you can add more samples if the model doesn't recognize your signs. If you add more samples, you need to train your model again.



## Did you notice?

Even when you don't hold any sign up, the model is trying to decide if that means "Drive" or "Stop" and is trying to put it in one of the classes. That will cause a lot of trouble!

## What can we do?

Let's make another class that means "No Signal". This will work for all the times we don't show anything.

### ▶ Make a "No Signal" class

Add a third class called "No Signal". For samples, show yourself without holding up a sign, or just your background. Once done, train and test your model again. If you're happy with it, move on to the next step: exporting your model.

### ▶ Export the model

- Click on "Export Model".
- Select "Upload (shareable link)" and wait until the upload is complete. Don't worry, it will not upload your images - just the computer model.
- Copy the link. Note: you can always come back and click on export model if you need to copy the link again. The address will not change even if you have to update your model.

### ▶ Done with the model!

Go back to intelino Scratch, but don't close the Teachable Machine!





## Now we will do the coding!

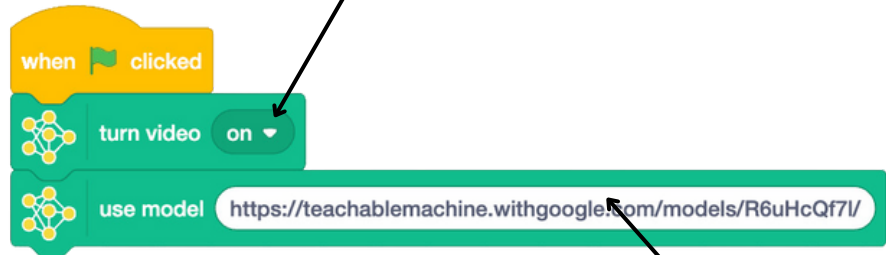
Let's start with this program:

### ▶ Make the program

Use File -> Load from your computer and open the file **"ai-sign-train.sb3"** or assemble it from the blocks so it looks like this program below.

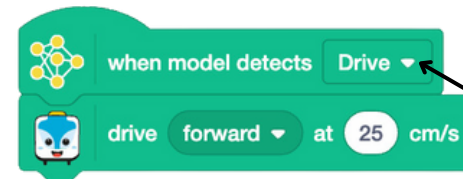
Study the blocks!

These blocks just start the program, turn on the camera, and load your model



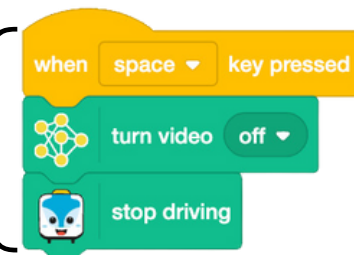
You can change the options for many of the blocks if needed

This is the event and response for driving



Paste the address of your model here

The camera and train do not automatically turn off when we're done running our program so these blocks make it easy: just press any key on the keyboard!



If you don't see the class Drive as an option, then Scratch doesn't know about your model yet. Simply start the program once by clicking on the green flag in the first block. Your classes should appear as an option now.



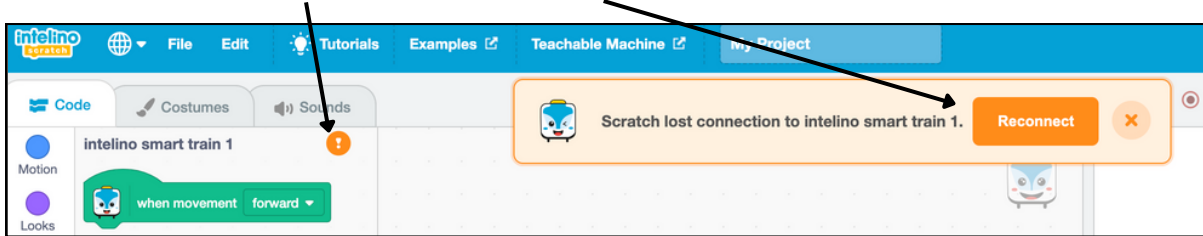
## Ok, let's run the program!

These are the steps:

### ▶ Make sure that the train is still connected

After a while, the train may turn off or disconnect to save energy.

Check if you see this orange circle or this box.



If so, do this:

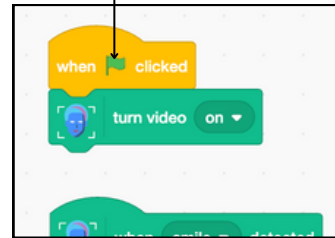
- make sure the engine is turned on, and
- click on either the orange circle or the Reconnect button and connect again like before.

### ▶ Place the engine anywhere on the track



### ▶ Run the program!

Click on one of the green flags!



### ▶ Give the sign

Hold the drive sign up to the webcam. Does the train start to drive?



**Tip:** Move the sign up to the camera quickly. Same when you put the sign down. The model can get confused when it only sees part of the sign if you move it in slowly.

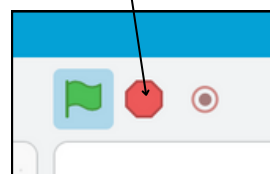
### Tip: how to stop a program



Click on the red flag

**AND**

Press any key to stop the engine and camera, or just pick up the engine and quick press the button.





## How about stopping at stations?

Now it's your turn to program the rest.

### ▶ Program Jamy so that it can stop at every station it comes across

Your code already covers the driving, now program an event that makes the train stop when the model recognizes the stop sign. That's the one with the red circle.



#### Add to your program

Many things don't have to change, just add the blocks you need.



#### To duplicate a set of blocks:

Right-click (on a computer) or long-press (on a tablet)



#### Refine your model

If your signs don't get recognized, you may need to go back to the teachable Machine and add more samples to your model. For example, have another person hold up the signs when recording sample images. Or change the background or lighting. The more relevant information the model has, the better it gets. Don't forget to retrain your model, test it out, and when you are happy with the results, update your model. The url will stay the same so you don't need to change that in the scratch program.



#### Divide the work!

It is really hard to watch the train AND hold up the signs at the same time. Have one person watch the train and shout out when to stop. Another person can work with the signs.



#### Debug: change and try again

If something doesn't work:

- stop running the program
- fix the commands (blocks)
- try again

This is called  
"debugging"

# Challenge **Take a Turn**

Difficulty ●●○



## Here's a challenge for you:

You may have noticed that I turn or go straight randomly. Add sign recognition to control my steering! This way, you can make sure that I stop at every station.

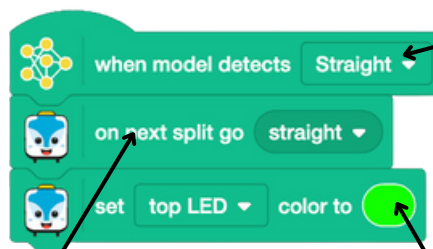
## ▶ Program Jamy so that

- Jamy drives at a speed of 25 cm/s.
- Jamy stops at every station at least once.
- Jamy can steer at split tracks.

## Here are the steps:

- Add classes to your model that recognize the other sign cards you have for going straight, left, and right.
- In Scratch, add events to control Jamy at the next split track (go straight, turn left, turn right).
- When Jamy is driving, control the train so that it stops at every station. Don't skip any!

## This is an example of what you would add:



Note: if you happen to give more than one direction command before Jamy reaches the split, then they will overwrite each other.

What does that mean? Imagine, for example, that you tell Jamy to go straight, but then give a turn command before Jamy reaches the split. Then Jamy will drive according to the last command you have given. In this case, Jamy would turn.

This makes Jamy go straight. But make sure you are choosing the right direction for the turn - it could be left or right depending on the way you let Jamy drive!

Tip: setting a different colors for different steering directions makes it easier to see which one is set for the next split.



## We made our first AI model!

But I still have some questions:

### Why don't we just use humans for some of these tasks?

AI models can be our assistants for some task that would take us too long or tasks that can be automated. AI models have become so good that they are sometimes much faster, better, and more reliable. And one model can be used in many places at the same time - all over the world!

In real life, AI models have made so many task much easier for us. For example, models that can read street signs can make sure that trains do not miss any of the sign. This makes the train ride much more safe!

### Is AI scary?

New things can be scary if we don't know what they can and cannot do and if we don't know how they work. This is why we started to learn about AI today!

With knowledge and rules about how to use AI, we can make sure that it is safe for everyone!

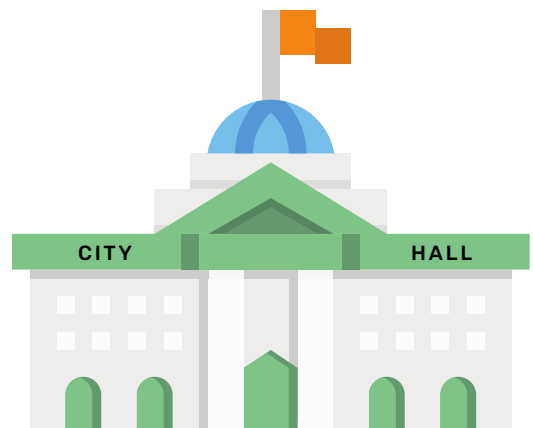


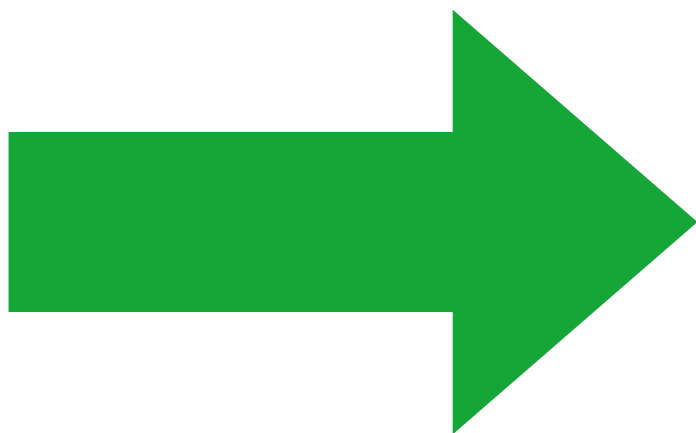
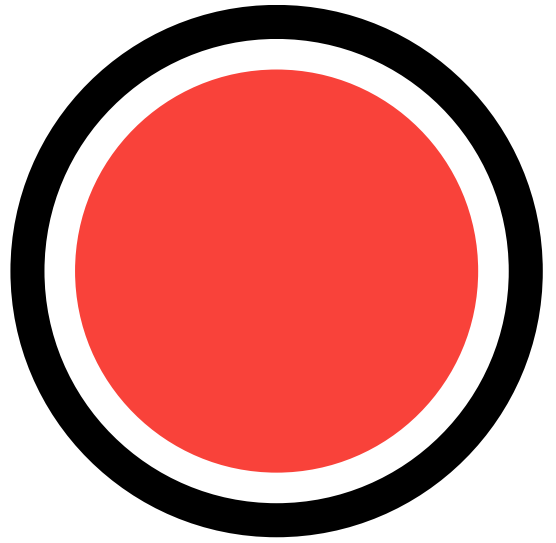
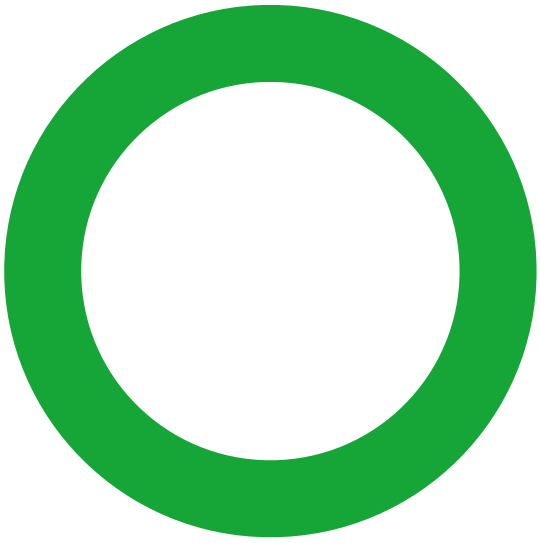
## Do you want to explore more?

Here are some ideas:


### ▶ Some ideas to try

- Make your own sign and control more actions like reverse direction, slowing down, or speeding up.
- Try out other Teachable Machine models like Sounds or Poses.





# Solutions

when clicked  page 10

turn video on

use model <https://teachablemachine.withgoogle.com/models/R6uHcQf7I/>

when model detects Drive

drive forward at 25 cm/s

when model detects Stop

stop driving

when space key pressed

turn video off

stop driving

when clicked **Challenge**

turn video on

use model <https://teachablemachine.withgoogle.com/models/SpHRf5rJX/>

when model detects Drive


drive forward at 25 cm/s

when model detects Stop

stop driving

when model detects Straight

on next split go straight

set top LED color to 

when model detects Left

on next split go left

set top LED color to 

when space key pressed

turn video off

stop driving

when model detects Right

on next split go right

set top LED color to 