cubelets

ROBOT BLOCKS

Getting Started Guide

Modular Robotics
1860 38th St.
Boulder, CO 80301

www.modrobotics.com

1. Make Your First Robot

The Dimbot – Uses a clear Flashlight Action block, black Distance Sense block, and a blueish-gray Battery block. It doesn't matter where you put the Battery block.

When you snap together a Battery block, a Distance Sense block, and a Flashlight Action block, the Flashlight Action block lights up. You control its brightness by moving your hand or an object closer or further from the "eyes" on the black Distance block.

This robot's light dims when you move your hand away, so you could call it a Dimbot.

2. Understanding Your Cubelets

Cubelets come in three types: **Sense blocks, Action blocks, and Think blocks**. Sense blocks are black, Action blocks are clear, and Think blocks are different colors. Most Cubelets have five connection faces and one special face, which identifies the function of that Cubelet. Others have six connection faces and their function is indicated by their color.

Every Cubelet has a small LED light in one corner. When the Cubelet is part of a robot and the robot's Battery block is turned on, the LED light is on, too. The LED light shows that the Cubelet is getting power and talking to its neighbors. Each Cubelet robot must have one Battery block which powers all the other blocks in the robot.

The Battery block has a small switch. When you slide it to the "O," the Battery block is off. Slide to the line, "I," and it is on. Turn it off to save battery life when you aren't playing with your Cubelets.

The Battery block has an internal, rechargeable battery. To recharge the Battery block, plug it into a micro-USB power source.

Each connecting face of a Cubelet has three conductors. The outer ring and magnets conduct ground; the inner metal ring conducts power; and the center pin conducts data from one Cubelet to the next. These three conductors must connect with their neighboring counterparts in order for two Cubelets to communicate.

3. What's a Robot?

A robot is a machine that senses its surroundings and acts on its surroundings. Every robot needs a Sense block and an Action block. Sense blocks are black and Action blocks are clear.

Every robot needs power. The blue-gray block is the Battery block. The Battery block has an on-off switch. Make sure it's on before you start to play; turn it off when you're done.

To build a robot, you will need a gray block, a black block, and a clear block. Just snap them together and you've built a robot!

4. Swap Sense Blocks

Now take out the Dimbot's Distance Sense block. Put a Brightness Sense block in its place. You still have a flashlight robot, but now its brightness depends on the light around it.

Test it: Cover the Brightness Sense block with your hand and the flashlight dims. Move your hand away and the Flashlight gets brighter.

These cubes are modular which means you can swap any Sense block for any other Sense block. You'll still have a robot, just a different robot. In the Dimbot we swapped a Distance Sense block for a Brightness Sense block to make a Light-sensitive Dimbot.

Accelerate your learning with a quick and easy getting started video, watch it on YouTube.

https://youtu.be/YPAOCOJibfQ

This guide is also available on the Modular Robotics website!

http://www.modrobotics.com/cubelets/cubelets-getting-started/

5. Swap Action Blocks

This simple brightness go-bot robot has a Drive Action block (left) that moves the robot when its Brightness Sense block sees light. The gray block on the right is the Battery block.

You can also swap Action blocks. With a Light-sensitive Dimbot now put a Drive Action block in place of the Flashlight Action block. Now your robot has a Brightness Sense and a Drive block (and, of course, a Battery block).

This robot moves when it senses light. In a bright room it's a fast robot. In a dark room it's a Slowbot.

Try more swapping.

What if you use a Speaker Action block instead of the Drive Action block? You get a Canarybot.

What about a Distance Sense block instead of a Brightness Sense block? You get a Fraidybot or a Friendlybot depending on the way the wheels are positioned. By turning the Drive Action block so that the robot goes backward instead of forward, you'll have a Friendlybot or Fraidybot.

6. How Numbers Flow

The arrow shows the flow of a number from the Brightness Sense block to the Flashlight Action block.

Each black Sense block senses some property of its surroundings and turns it into a number. Each Sense block tells its number to all its neighbors. You can see them “talking” as the green lights on each block flicker. (The Bluetooth flashes different colors.)

For example, the Knob Sense block senses how much you rotate its knob. When you turn the knob all the way counterclockwise (left), the Knob Sense block produces a small number. Turn it clockwise, to the right, to produce a big number.

The Brightness Sense block senses how light the room is. In a dark place, the Brightness Sense block produces a small number. In a light place, the Brightness Sense block produces a big number.

Each Action block takes numbers from its neighbors and turns the numbers into an action.

The Flashlight Action block takes a number and lights its lamp. A big number makes the lamp bright. A small number makes the lamp dim. Think of the number as hopping or flowing from one block to the next. Numbers are flowing through the blocks of the robot from Sense blocks to Action blocks all the time. That’s what makes the robot behave the way it does.

Numbers don’t flow through Sense blocks. Each Sense block produces its own number, so it doesn’t pass numbers from its neighbors.

7. Using the Bar Graph to See the Numbers

You can use the Bar Graph Action block to understand what's going on inside your robot—to show the numbers flowing from block to block.

Attach the Bar Graph Action block to any block in a robot. The Bar Graph Action block shows how big the number is. If the number is big, all the cells in the bar graph light. If the number is small, only a few light up. If the number is very small (or zero) no cells light at all.

Try it. Build a simple Brightness Gobot with a Brightness Sense block and a Drive Action block. Attach the Bar Graph Action block to one of the blocks. If there's a lot of light, the Bar Graph Action block will show a full bar (and the Drive Action block will move fast). If there's not much light, the Bar Graph Action block won't light much.

You don't need the Bar Graph Action block to understand what's going on with this simple robot. But, with bigger and more complicated robots, the Bar Graph Action block can help.

8. Arrangement of Cubelets Makes a Difference

The pictured robots have different versions because you can put the Drive Action block into the robot in different ways. One way, the robot goes straight. The other way, the robot goes around and around—you could call it a Turnabot.

It's not just which Sensor and Action blocks you choose. It's also how you position and situate the Cubelets. The same Cubelets arranged in a different physical configuration make different robots.

For example, make a robot and place the Brightness Sense block so its sensor face points down. Now it doesn't see the light. This Gobot is a Nogobot. No matter how bright the room is, the robot won't go. Its Brightness Sense block doesn't sense the light.

Try placing the Brightness Sense block to face in different directions. How does that impact the robot's behavior?

9. Stability

Some robots are more stable than others. These robots all have a Distance Sense block and a Drive Action block, but the differences in their assembly produce different behaviors.

Try building a simple Gobot with a Distance Sense block and a Drive Action block. It's stable if you build a train with all three blocks (the Battery block, the Drive Action block, and the Distance Sense block) arranged in a row.

The Distance Sense block produces a big number when something gets near it. You can chase this Fraidybot around with your hand. When you get near the Fraidybot, it runs away.

If you build this robot as a tower instead of a train, it still works, but it's no longer stable: approach the Distance Sense block and the Drive Action block starts moving. Accelerate too quickly and the tower falls over.

You can fix this: add a block at the bottom next to the Drive Action block. Any block will do, but try one of the green blocks, either a Passive block or a Blocker block.

Notice that you can build this Gobot in different ways. If the Distance Sense block faces the same direction that the Drive Action block moves, your robot comes towards your hand.

If you turn the Distance Sense so it faces the opposite direction, it moves away. You can chase it around with your hand or change the direction of the wheels on the Drive Action Block.

10. A Sense Can Control Multiple Action Blocks

You can use a single Sense block to control one or more Action blocks.

Build a Lighthousebot that uses the Knob Sense block to control the speed and the brightness of a rotating robot tower. The Lighthousebot uses the Knob Sense block to control the speed of the Rotate Action block and the brightness of the Flashlight Action block.

Build a simple Gobot with a Brightness Sense block and a Drive Action block: It goes when it senses light.

Add a Speaker Action block. Now it goes and chirps when it senses light. Add a Flashlight Action block. Now it goes and chirps and lights up when it senses light. Add all the Action blocks you want. They all respond to the same Brightness Sense block.

With a lot of light on the Brightness Sense block, all the Action blocks will act a lot. Without light on the Brightness Sense block, the Action blocks won't do much.

11. Think Blocks

You've met the Battery block, and the black Sense blocks and the clear Action blocks. It's time to meet the colored Think blocks. Think blocks are the colored blocks.

Because Robots are machines that sense first, then think, then act, we need to be sure Think blocks are placed between the Sense block and the Action block you want to impact.

In addition to the green Passive block, the simplest Think block is the red Inverse Think block.

12. The Inverse Think Block

To make a Nightbot light up when it's in a dark place, you need a Red Inverse block. Ooops! This robot as pictured is wrong. Can you figure out why?

Let's go back to the Light-sensitive Dimbot. It has a Brightness Sense block and a Flashlight Action block. (It also has a Battery block, of course, but we're going to stop mentioning the Battery block because every robot has one.)

The Dimbot made a silly flashlight. Its lamp is bright when it's in a bright room, and dark when it's in a dark room. We'd prefer a flashlight robot that turns on when it's dark, and turns off when it's light. That's why we need the red Inverse Think block.

Put the red Inverse block between the Brightness Sense block and the Flashlight Action block. Remember the number flow story: every Sense block produces a number. The Brightness Sense block produces a big number when it senses a lot of light. It tells that big number to its neighbor (the Flashlight Action block), which turns the big number into a bright light.

The Inverse Think block turns a big number into a small number (and a small number into a big number).

When we put the Inverse Think block into the robot, numbers pass through it from Sense block to Action block. When the Brightness Sense block senses a lot of light it produces a big number, which the Inverse Think block turns into a small number, and passes to the Flashlight Action block, which dims its lamp.

It also works the other way. When you put the robot in a dark room (or shade it with your hand) the Brightness Sense block produces a small number. The Inverse Think block turns it into a big number, and the Flashlight Action block makes its light bright: A Nightbot that turns on in a dark room, and turns off in a bright room.

Swap out the Flashlight Action block and put a Drive Action block in its place. Now you've built a robot that goes when it's dark, and stops when it's light.

The Inverse Think block in this Night-Gobot inverts the number from the Brightness Sense block before passing it to the Drive Action block. In low light, the Brightness Sense block produces a small number; the Inverse Think block inverts it to a big number, which makes the Drive Action block go fast. In bright light, the Brightness Sense block produces a big number, which the Inverse Think block inverts to a small number, so the Drive Action block moves slowly, or not at all.

13. Differential Drive

Put two Gobots together on a robot (facing the same way) and you've built a Steeringbot. The Steeringbot has two Gobot towers with a Battery block in between. Each Gobot tower responds to nearby objects. Put your hand near the right side Gobot tower and its Drive Action block will go, while the left side's Drive Action block stays still (or goes slower).

When one Drive Action block moves and the other doesn't, or turns the other way, the Steeringbot turns. That's called "differential drive steering."

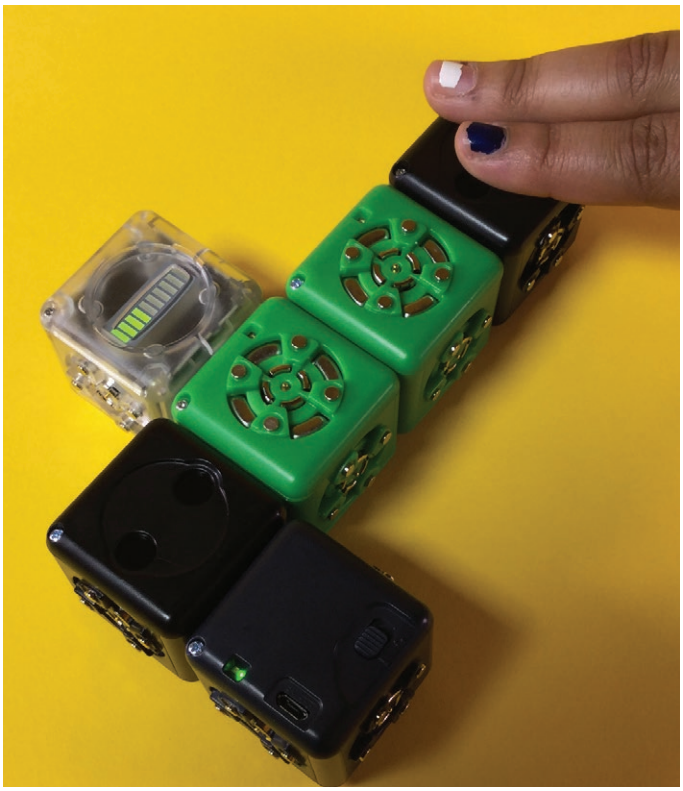
14. Action Blocks Average Their Inputs

If your robot has two Sense blocks and one Action block, which one controls the robot? Build a Testbot with two Distance Sense blocks and a Bar Graph Action block between them.

The Bar Graph Action block shows a low value if neither Distance Sense block senses an object. Put one hand in front of each Distance Sense block. They will both produce a high number. The Bar Graph Action block shows a high number.

Now put your hand in front of just one of the Distance Sense blocks. This block now produces a high number while the other Distance Sense block produces a low number. The Bar Graph Action block takes both numbers and averages them. It shows a number that is halfway between the numbers it gets from its two Distance Sense neighbors.

15. Gradients: Diffusion

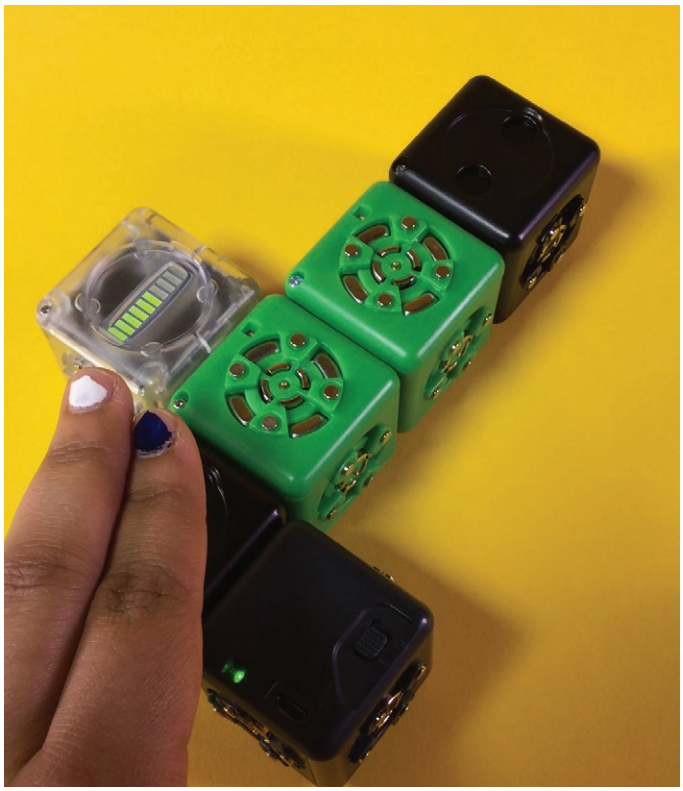


This robot shows the gradient story. The Bar Graph Action block is attached to one of two Passive blocks between two Distance Sense blocks, one at each end of the robot. The closer Distance Sense block has a bigger effect on the Bar Graph Action block. If your robot has an Action block right between two Sense blocks, the Action block averages the numbers the two Sense blocks tell it.

If one Sense block is farther from the Action block than the other, then the closer block has a stronger effect. You can test this using the Bar Graph block. Build a robot with the two Distance Sense blocks on either end and two Passive blocks between them. Add a Bar Graph Action block to one of the Passive blocks.

Now play with your robot: put one hand in front of each Distance Sense block, so that the Bar Graph Action block reads high (all its cells light). Take your hand away from the Distance Sense block that is farther from the Bar Graph Action block. Now put your hand back and try the other Distance Sense block. The Bar Graph Action block responds more strongly to the closer Distance Sense block.

An Action block acts according to the average of the Sense block numbers it gets, weighted by the distances (number of blocks from Sense block to Action block, or “hop count”) they’ve travelled.



16. Use The Minimum Block as a Switch

Suppose you want to make a light-sensitive Gobot—it goes when it sees light. But you also want to be able to turn it off. Of course, you can just switch off the battery, or take out the Battery block. But you can also use Cubelets to make an on-off switch. Here's how.

An ordinary Light-sensitive Gobot just has two blocks: a Brightness Sense block and a Drive Action block (plus a Battery block). The number from the Brightness Sense block tells the Drive Action block how fast to go. A brighter light, a bigger number, a faster go-bot.

Take out the Brightness Sense block and put a Minimum Think block in its place. This Minimum Think block will tell the Drive Action block how fast to go. It will take all the numbers its Sense block neighbors give it, and choose the smallest (minimum) of those numbers. This smallest number is what it will pass on to its Action block neighbors.

Attach the Brightness Sense block to the Minimum Think block, and also attach a Knob Sense block. Now, if you turn the Knob Sense block all the way counterclockwise (left), then the Minimum Think block tells the Drive Action block, “zero”, because this is the smaller of the numbers it’s getting. If you turn the Knob Sense block all the way clockwise (right), then the Minimum Think block will tell the Drive Action block whatever number it’s getting from the Brightness Sense block. With the Minimum Think block, the Knob Sense block acts like an on-off switch.

17. Use the Blocker Block to Separate Two Parts of a Robot

The dark green Blocker Think block passes power but does not allow numbers to flow through it. Use it to build a robot with two parts that don’t talk to each other. Here’s an example: One half of the robot is a Lighthousebot with a spinning light; the other half is a robot that chirps when it sees the light from the Lighthousebot.

The Lighthousebot is Knob Sense block that controls the speed of a Rotate Action block, and on top of that, a Flashlight Action block that points outward (sideways). When you turn the Knob Sense clockwise (to the right), the light goes on and spins.

Now add a Blocker block to the base (say, on the Battery). Then, on the other side of the Blocker block add a Speaker Action block, and, on top, put a Brightness Sense block with its sensor face pointing towards the rotating light. The Speaker Action block chirps (if the room is bright).

To the right of the green Blocker block is a mini lighthouse: a Knob Sense block, a Rotate Action block and a Flashlight Action block. The Flashlight Action block spins when you turn up the Knob Sense block. When light from the spinning Flashlight Action block strikes the Brightness Sense block, the Speaker Action block responds by chirping faster.

18. Using the Threshold

The Threshold Cubelet is a Think block with an adjustable knob. Unlike most Think Cubelets, the Threshold allows you to alter its effect within the robot. The Threshold Cubelet acts as an insulator, or blocker, of numbers below a value established by the Threshold knob. Numbers above the set Threshold value will continue to flow from block to block.

Construct a Testbot with a Battery, Bar Graph, and Distance Sensor. The closer an object is to the Distance Sensor, the higher the value displayed on the Bar Graph Cubelet. Now, place the Threshold Cubelet between the Distance Sensor and Bar Graph. Set the Threshold value to the lowest setting. You'll notice that the Threshold Cubelet doesn't seem to have any effect. That's because we've set the Threshold value at a low number... so, any number above the Threshold value, 0, will pass on from block to block.

If you turn the Threshold Knob up to about 50 percent, you'll notice that numbers below the halfway mark don't appear on the Bar Graph display. Once numbers from the Distance Sensor exceed the Threshold value, you'll notice that the Bar Graph display will show values normally.

You can adjust the Threshold Cubelet settings to create robots that respond suddenly or to build binary behaviors.

19. Using the Bluetooth

The Bluetooth Cubelet is a THINK block with special abilities. It has a Bluetooth radio inside that enables devices like a computer, tablet, or smartphone to communicate with Cubelets. Using the FREE Cubelets App you can control your robots with any iOS or Android device. With the release of Cubelets OS 4, Modular Robotics periodically releases firmware updates and new features via the Bluetooth Cubelet.

http://www.modrobotics.com/cubelets/bluetooth-getting-started/

http://www.modrobotics.com/cubelets/apps/cubelets-flash/

20. A Note to Experienced Programmers

Experienced programmers often ask, “So which block is the IF-THEN block”. Or, “Which block is the CPU?” We understand the questions, but that’s not how Cubelets work.

Building robots with Cubelets is different from the procedural programming (in C, Java, or BASIC) that you may know. In procedural programming, a robot’s behavior results from executing a sequence of instructions in the robot’s “brain” (usually a single microcontroller).

Cubelets operate with a completely different model: distributed programming.

Every Cubelet has a microcontroller. The robot’s behavior results from local interactions between Sense, Think, and Action blocks and the numbers flowing from block to block.

There’s no single “brain” block, and there’s no sequence of instructions. There are no variables, functions, or procedural logic. Instead, in Cubelets, the robot is the program. The way you put the blocks together determines the way numbers flow from Sense to Action blocks, and this determines your robot’s behavior.

Need help? Email us!
support@modrobotics.com