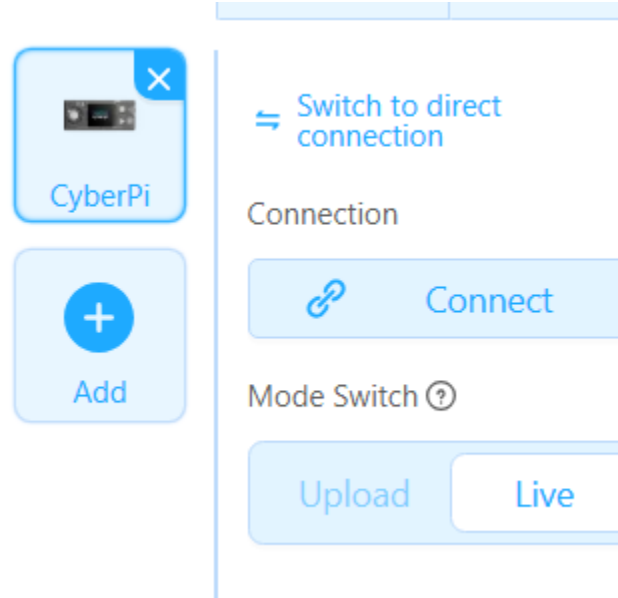














mBot Adventures: Workshop Instructions

- I. Before we start the first part of our workshop, we need to make sure all the bots have the firmware reset so that the bots have a clean slate when coding them
 - a. First plug in the USB cable into the mBot mega
 - b. Open the mBlock IDE on your laptop using this link: <https://ide.makeblock.com/>
 - c. Make sure that the mBot mega is added to your devices area




Search Device Library

☒ All
 ☐ Official development
 ☐ Non-official development


 <p>mBot2 Developers: By mBlock o...</p>	 <p>mBot2 探索者 Developers: By mBlock o...</p>	 <p>CyberPi Developers: By mBlock o...</p>	 <p>xLight Developers: By mBlock o...</p>	 <p>mBot Mega Developers: By mBlock o...</p>	 <p>Halocode Developers: By mBlock o...</p>
 <p>Codev</p>	 <p>mBot</p>	 <p>mBot Ranger</p>	 <p>Ultimate 2.0</p>	 <p>mBuild</p>	 <p>NovaPi</p>

Become a developer of mBlock to unlock more potential.


Devices



CyberPi



mBot Me...




Add

Sprites

Background


Switch to direct connection

Connection

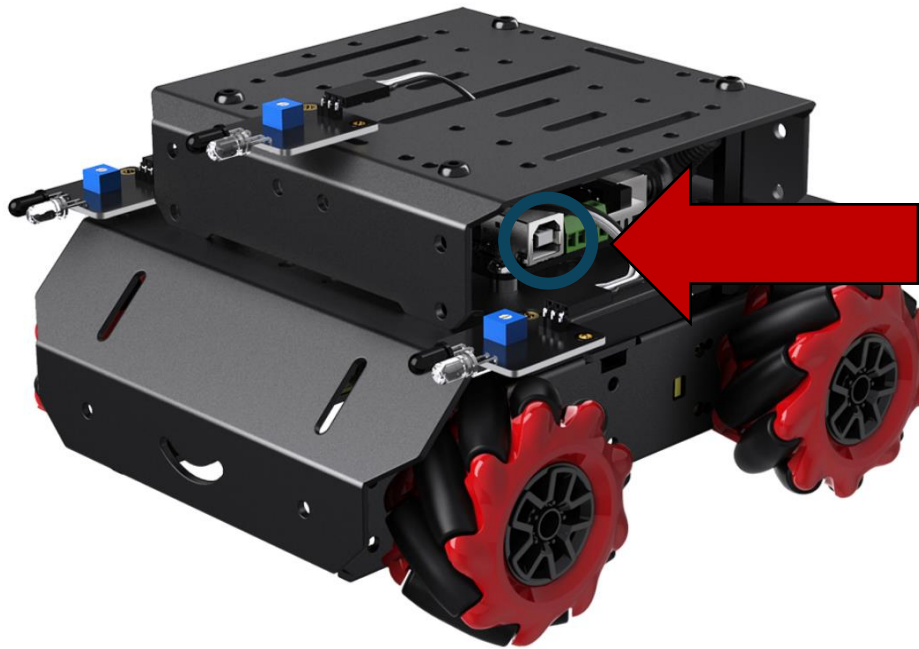

Connect

Mode Switch ?

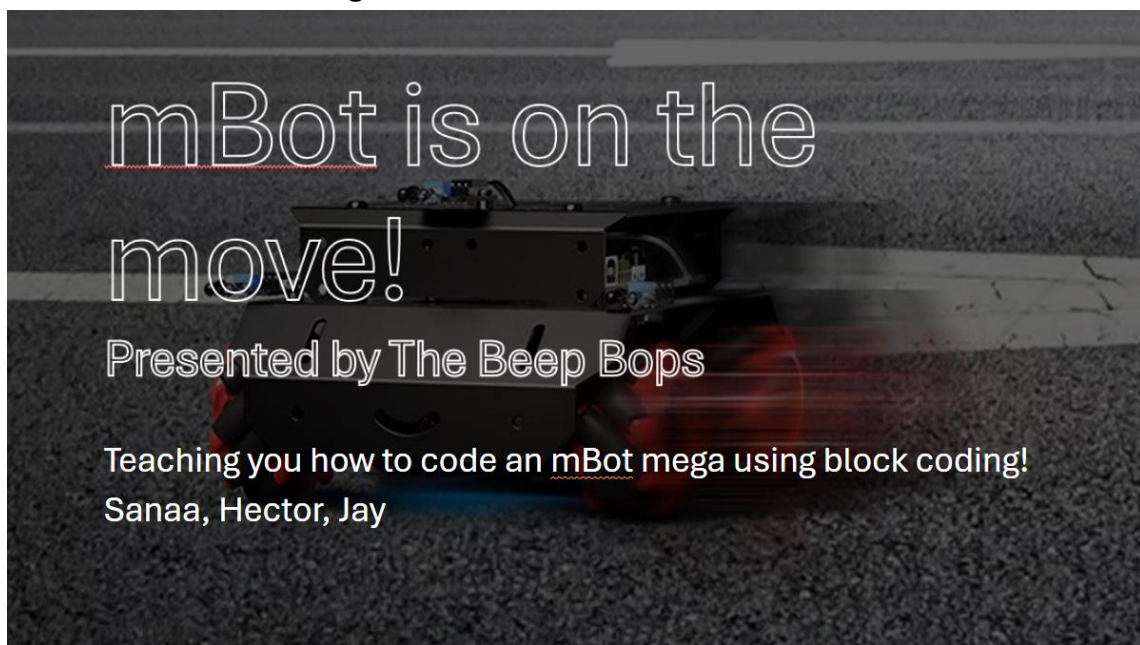
Upload
Live



- d. After it's been added, you have to click the connect button and establish a connection to the bot



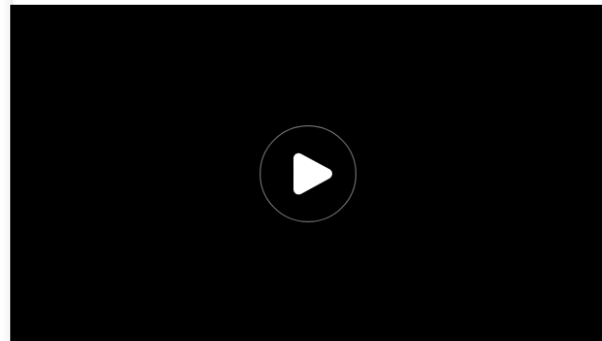
- II. Once the firmware for the bots has been updated and the class is situated, we will begin the lecture
- III. For the first slide we will introduce our group name, The Beep Bops, and then introduce ourselves along with our title name afterwards.



- IV. On our second slide, we show the class our commercial video which gives a short demo of what the bot can do and some of its functions.

Introduction

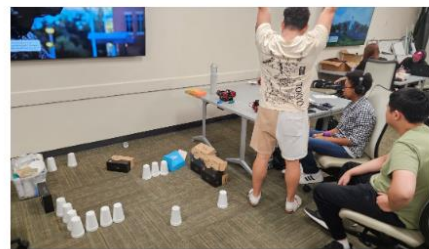
- We're the Beep Bops! This is a little video to show what we do.



- V. On our third slide, we give an explanation of what the TAP Program is. We give other details as well that aren't on the slide like how what we do would be good to put on your resume, i.e., our workshops, and how jobs would ask about what we did during these workshops.

What is TAP?

- Technology Ambassador Program (TAP) is a project-based class that provides a collaborative environment for students to work with their fellow classmates on a semester-long project using technologies of their choice. TAP strives to increase participation in IT through numerous outreach activities and workshops that are designed to showcase the creative and fun side of technology.



- VI. On our fourth slide, we explain what we want to teach in this workshop. We want to teach the students about block coding and how to apply it to the mBot. We hope to show the students what it is like to work in the field on a very small scale and show them that regardless of any knowledge, anyone can learn and do this type of work themselves.

Our teaching goal

Our main teaching goal will be teaching you to create block coding, learning if statements and while loops, and apply everything covered to the mBot.

This study is geared towards anyone interested in coding, or getting to know more about what it's like in the tech field

We're doing this to gain more engagement in the technology world and teach people that anyone can do this kind of work

- VII. For our fifth slide, we just read it from the slide. This slide is basically an explanation of what we will go through in the workshop.

What you will be learning today !

- What a program is
- What is an mBot?
- What block coding is
- If statements, Forever loops, and Logical Operations
- Three Stages to coding the bot:
 - 1. Programming it to go back and forth
 - 2. Programming it use the sensors to rotate
 - 3. Programming the crash detection

- VIII. The sixth slide is also essentially the same by just reading from the slide. This slide gives an explanation of all the functions of the mBot mega. We also gave a demonstration in person by showing everyone in the class where all the sensors were on the robot.

What is mBot?

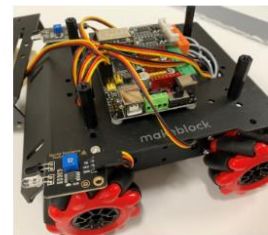
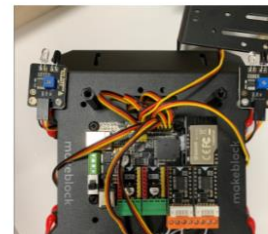
- The mBot smart control robot that can be controlled either through block coding or through a controller
 - There is an app that includes a controller that can be connected through Bluetooth on the mBot
- The mBot has many different functions
 - Line Detection Sensor
 - Crash Detection Sensors
 - Motion Sensors
 - LED Lights
 - 360 Rotation



- IX. The seventh slide is an extension of the explanation for the mBot mega. It gives a more in-depth explanation, along with pictures, of what the raspberry pi is on the mBot mega and how it functions.

More about mBot Mega...

- mBot Mega essentially uses a Raspberry Pi, a small computer, as the hub for which everything connects to
- The Raspberry Pi, called the MegaPi for mBot, allows the bot to have bluetooth connection
 - This is where we will be connecting our bot to our computers to upload the code
 - There is also a switch that turns the robot on and off



- X. On the eighth slide, before we go into the explanation, we ask the class what do they think when they think of someone who programs? After their answers we go into saying how there is usually a stereotype of someone who sits at their desk all day and codes. We then go onto reading the slide and explaining how everyday programs they use, like Google Chrome and Firefox, are made by using coding. We

also explain how the computers are never wrong when they code. If there is something wrong with your code it is due to human error.

Programming ?

- What is even a program ?
 - A sequence of coded instructions for a computer or machine to perform
 - Basically, telling a computer we want it to do!
- Examples ?
 - Operating systems like MacOS or Windows
 - Word, excel
 - Google chrome, Firefox

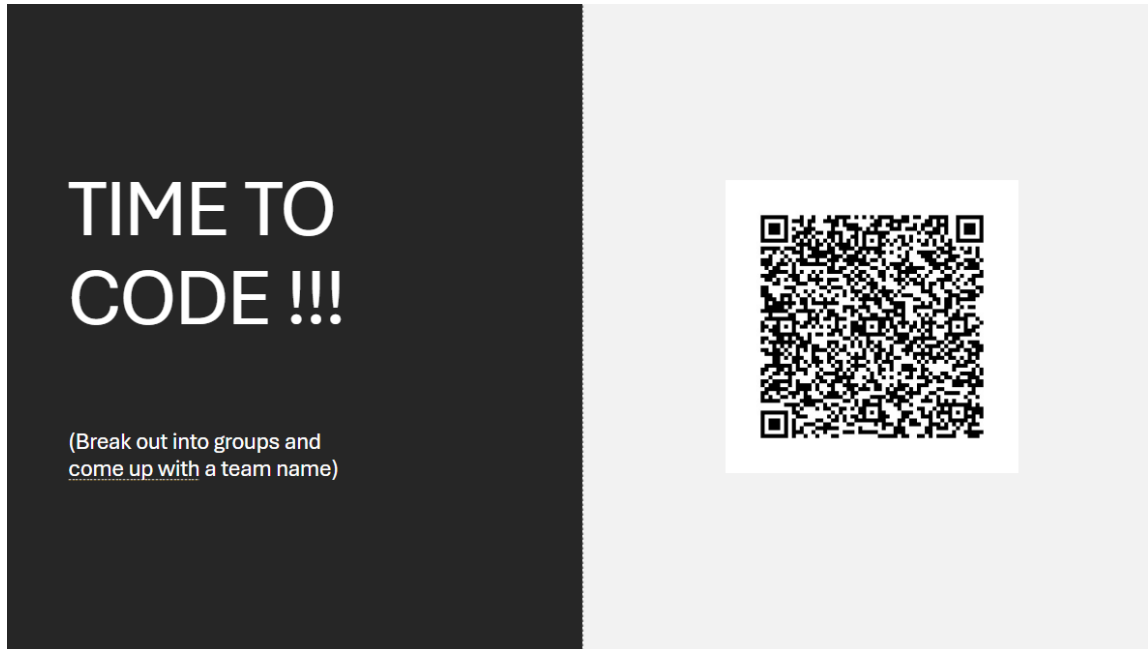
XI. On the ninth slide, we mainly read directly from the slide just giving a short explanation on what block coding is. We also mention how this is the first step to take when learning programming, starting with block coding. We also show in the picture examples of how the colors correlate to different meanings in block coding, dark blue for motion, light blue for sensing, and green for operators.

What is block coding?

- Block coding is dragging blocks of code together like a puzzle to create a program
 - These blocks can range from telling the bot to move forward, backwards, turn around, or even turn on and off its LED lights
- The different colors for the blocks also have different meanings
 - In mBlock, orange is for control, yellow is for events, dark blue is for motion, etc.
- This is an alternative way to code instead of using languages like JavaScript or Python

XII. On the tenth slide, we tell the students to form groups and start to hand out the mBots to them. We also ask them to scan the QR code on the screen. This QR code

gives them a cheat sheet in case they cannot follow along when we code the bot with them.



- XIII. The next three slides go into detail on how to connect the bot to the computer
 - a. Tell them to go to the link <https://ide.mblock.cc>
 - b. Next have them add the mBot to the devices like how we did before when resetting the firmware for the bot
 - c. Once added make sure the mode switch is on upload
 - d. Afterwards, plug in the bot through the raspberry pi using the USB cord to your computer
 - e. Then make sure to hit connect to finally connect the bot to your device
- XIV. On the 14th slide, this is when we start to go over the first part of the code. We show them how to code the mBot to move back and forth. After opening the IDE, you can follow along the cheat sheet yourself to teach the class. When we did this, we also had other people in our team go around the class and help those that struggled to follow along. When we finish coding, we tell the students to upload the code to the bot and let it run a demonstration on the floor.
- XV. On the 15th slide, we go into an explanation of what an if statement is. We mainly read from the slide here, but we also give a real-life example. To stop a car, you must press the brakes.

If statements

- Is a conditional statement
- A nested if statement allows you to test multiple different conditions increasing the number of outcomes (We will have these in some of the code we will program)
 - Allows us to control the flow of a program
 - If the condition is true, it runs the statement inside the block
 - If the statement is false, it ignores the block and moves onto the next command

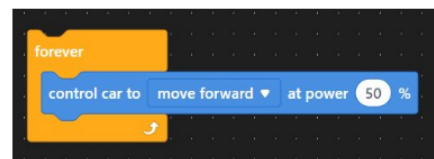


In the image above, it shows that if the sensor detects an obstacle the bot will stop moving, and if it does not detect anything the bot will move forward at a power of 50%

- XVI. On the 16th slide, we go over what a forever loop is. We go on to explain that as the name states, it is a loop that never stops. A real-life example we gave was how if you don't tell a car to stop moving or press the brakes, it will continue to move forward.

Forever loops

- Is a type of loop that goes on forever
 - There is no limit to how many times it can repeat
 - There are two other types of loops as well, for and while loops, but the one we will be using is a forever loop



In the picture above, it shows that the bot will continue to move forward at a power of 50%

- XVII. On the 17th slide, we go over what logical operations are. We mainly read from the slide for this one. But a real-life example we use for this is to enter the gym. When you need both your GGC ID and student number, you can get inside. For an OR statement, you can have either or and still get inside the gym.

Logical Operations

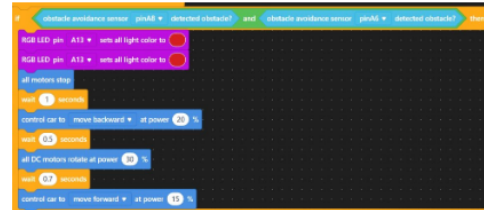
An operation is used to tell if two statements in a condition are true or false



There are two different operators AND and OR

A statement with an AND operator means that both conditions must be true

A statement with an OR operator means that either or condition must be true



An example from our crash detection code, if both sensors are activated at the same time the bot will move backwards

Logical operations use Booleans, which have the values of true or false

- These Booleans binary values are 0 and 1, 0 meaning false and 1 meaning true

- XVIII. On the 18th slide, we go into the next part of the code. We will code the bot to detect obstacles, back up, rotate, and then continue to move forward. This will be using if statements, forever loops, and logical operations. The second part of the code needed to follow along will also be found on the cheat sheet after the first part of code. We do the same procedure as when having the students do the first part of the code, we have other teammates go around helping, and once done, we have them upload the code and have them run on the floor. Some people could block it with their feet or hands, and others could even try it out in our obstacle course.
- XIX. On our 19th and final slide, we give the students free reign to add their own improvements to the code or make something completely new. There is a third example in the cheat sheet of a more complicated version of our code that they can also use to follow along. Once they finish the code and upload it, we have the students race in our obstacle course by having two students go against each other in order to reach the finish line which is the blue boxed area at the end.