

Answer Guide Explorer Workbook Activities

Activity 1 – Bricks, Blocks, and Gears

Names of parts:

Left Picture: 1 x 4 Beam

Left Center Picture: 2 x 4 Plate with Holes

Center Picture: Axle (Size can not be determined)

Right Center Picture: 24-tooth Crown (or Bevel) Gear

Right Picture: RCX

Activity 2 – Robot or Not

Object Identification

Upper-Left: Automobile – Machine (although it may contain computers to efficiently operate the engine)

Upper-Center: Gripper (Robotic Arm) – Robot

Upper-Right: Spraying Robot – Robot

Middle-Left: iPod – Computer

Middle-Center: Laptop – Computer

Middle-Right: Dante Volcano Sampling Robot – Robot

Bottom-Left: Clothes Dryer – Machine

Bottom-Right: Mobile Phone – Computer

Activity 3 – If Only I had a Brain

Label the Parts of the RCX

On the Left – Top to Bottom

Input Sensor Ports

View Button

Output Ports

On-Off Button

On the Right – Top to Bottom

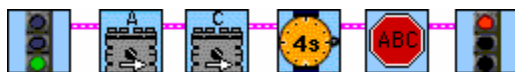
Infrared Port

Program Button

Display Window

Run Button

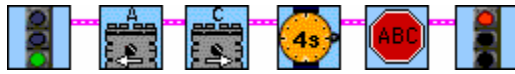
Activity 4 – Go Bot, Go



Activity 5 – Time and Time Again...With a Twist

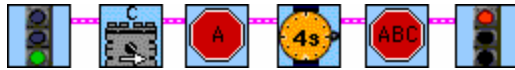
Three Ways to Make a Left Turn

Point Turn



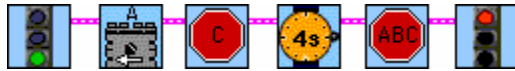
This looks almost like the program you wrote for your robot to go forward. The only thing you've changed is the direction of Motor 'A'. Now you have Motor 'A' going in reverse. This will cause your robot to turn on its own axis or make a 'point turn'.

Forward Non-Point Turn



Instead of running Motor 'A' in reverse, you've stopped it; and only Motor 'C' is running forward. Your robot will make a wide left turn, sort of like watching a cart make a left turn at an intersection.

Reverse Non-Point Turn

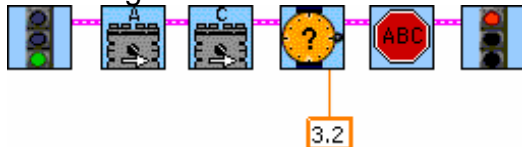


Here you have Motor 'A' running in reverse while Motor 'C' is stopped. You'll want to use this kind of turn whenever the front of your robot is up against a wall or some other obstruction and you want to make a turn. The next time you're in a car with your parents and they're backing out of a parking spot, watch how they turn the steering wheel. Since they're in reverse, if they want to go left, they'll turn the steering wheel right.

Activity 6 – An A-Mazing Experience

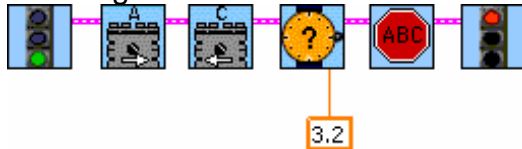
Everyone will design their maze differently, so it's important that you understand that all programs will be different – the important thing to look at is "Does it Work?" It doesn't matter how simple or complex the maze is, the program is only composed of three parts: go forward for a certain amount of time, turn left for a certain amount of time, and turn right for a certain amount of time.

Go Straight

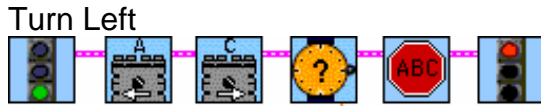


This program goes forward for 3.2 sec.

Turn Right



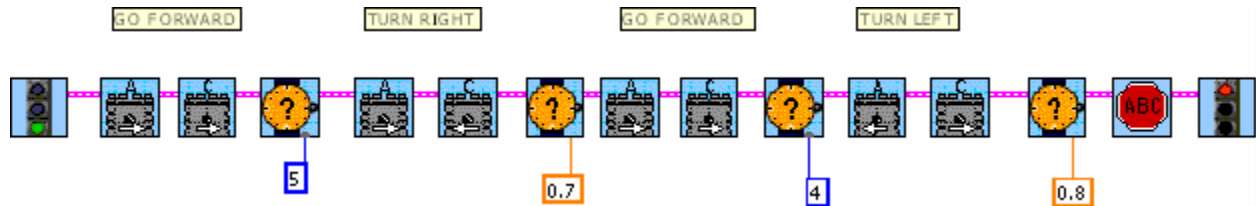
This program turns right for 3.2 seconds.





3.2

This program turns left for 3.2 seconds.

The next program ties all of these programs together to make your robot go forward, turn right, go forward again and then turn left. If your maze is more complex, you'll need to add more left or right or straight portions of the program.



Keep in mind that the modifiers will change from this example to your program, since your maze is probably different. Also, be very careful in making the point turns – you'll probably find that the amount of time needed to make a left turn is different than the amount of time needed to make a right turn. This is because of the difference in friction between the left and right wheels or treads.

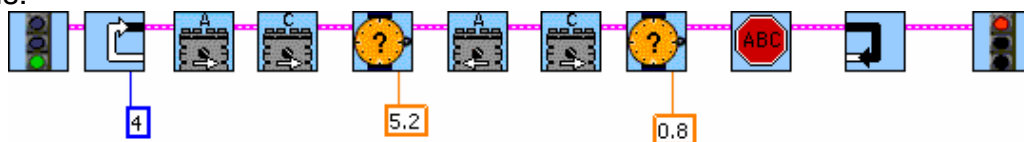
Finally, don't confuse  with . They look very much alike but the one on the right has a small letter 'm' in the lower right hand corner. The icon on the left is "Wait for Time" and the modifier is in seconds. The icon on the right is "Wait for Time (Min)" where the modifier is in minutes.

Activity 7 – Start Your Engines

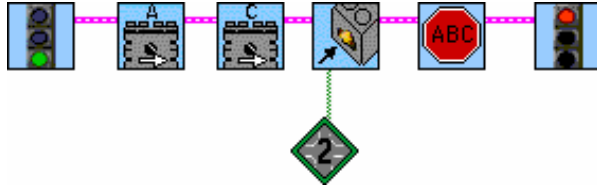
This is going to be just like what you did in the previous activity. The program is going to be based on Going Forward for a certain amount of time and then making a left point turn for a certain amount of time. Your basic program will look something like this (remember, the modifiers on your Wait for Time icons will probably be different):



Now you want your robot to go completely around the track. To do that, you'll need to run this program four times. You can right the basic program four times or you can use a loop. The Loop icons have a start and an end as well as a numeric modifier that tells the robot how many times to run the loop. In this case, you'll want to run the program 4 times to get around the four sides of the track. So your program should look like this:

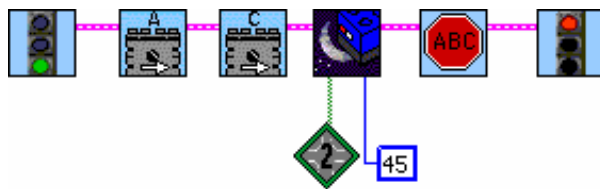


Activity 8 – When Push Comes to Shove





Don't forget to tell your robot where it can find the touch sensor. This program tells the robot to look for the touch sensor on Input Port 2. If you've connected your sensor to a different Input Port, be sure to put the correct Port Modifier in your program.

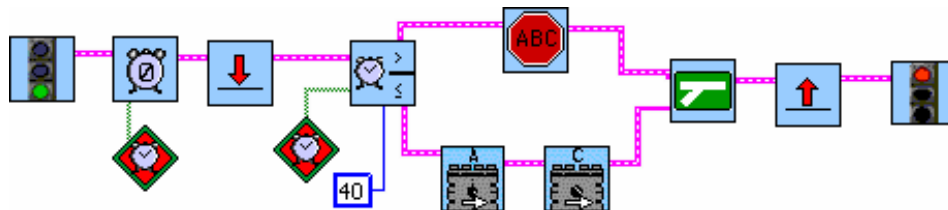
Activity 9 – Seeing the Light of Day



This program looks somewhat like the program you wrote in the last activity. The two differences are that you're using the "Wait for Dark" icon instead of the "Wait for Push" icon and you've had to include a threshold value for your sensor. Also, don't

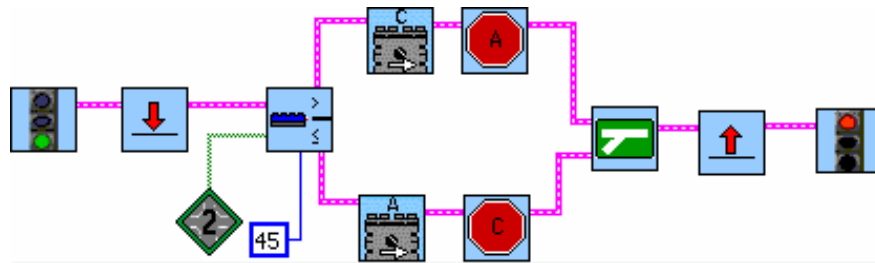
confuse  with . The icon on the left is "Wait for DARK", while the icon on the right is "Wait for DARKER". "Wait for Darker" waits for a value of brightness that is darker than the present value, which is the value the light sensor sees when the program comes upon the icon.

Activity 10 – Decisions, Decisions, Decisions



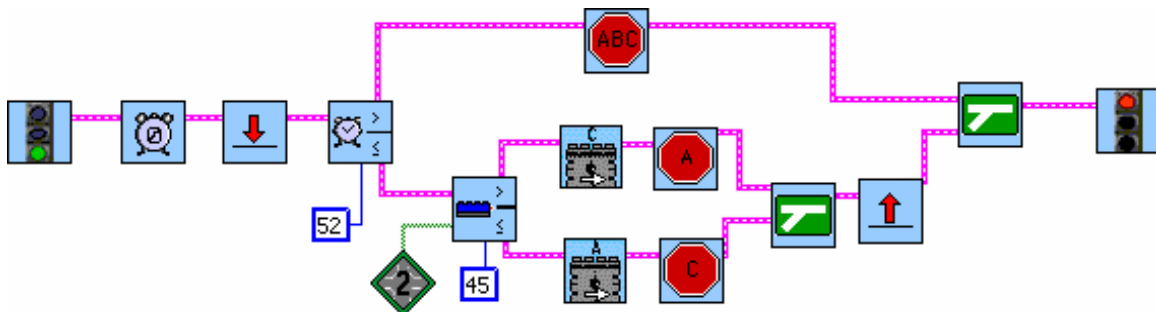
Besides the Timer Fork and the Red Jump and Land, there are a few more icons in this program that you haven't used before. The second icon (a clock with a zero in it) resets the clock. When you press the run button on your RCX, the clock begins to run. You want to reset this clock to zero just before you use it; if you don't reset the clock, the fork will begin at whatever time is already in the clock. You can use three different clocks at the same time – the clock modifiers are colors, red yellow and blue. If you're only going to be using one clock, AT A TIME, there is no need to put a modifier on the clock. Remember, you wanted to program your robot to go forward for four seconds, but the modifier for the timer is 40. That's because the time fork counts in tenths of a second and there are 40 tenths in four seconds.

Activity 11- Follow the Robot Road



Don't forget to tell your robot where your light sensor is attached. If you don't put a Port Modifier in your program, your RCX will look for the sensor on Port 1. Don't forget to tell your robot what the threshold value is. If you don't tell your robot what the threshold value is, it will think that the value is 55.

Activity 12 – Simon Says



In this program there are two forks, one for the timer and one for the light sensor. They are called "Nested Forks" because one fork is inside the other. Since you have two forks, you need to have two "Fork Merge" icons. This program will follow the line for 5.2 seconds and has a threshold of 45. You robot will probably follow the line for a different amount of time any may have a different threshold. Don't forget the Port Modifier for the light sensor fork so the robot knows where to look for the sensor.

