

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Team Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Grade \_\_\_\_\_\_ Lego Kit Number \_\_\_\_\_\_\_\_\_\_\_\_

Age \_\_\_\_\_\_\_\_

# At the beginning of each day…

1. Check out your team’s Lego kit
2. Search the area for Lego pieces

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked out by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Returned by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Learning Log

**At the end of each day, after you return your Lego Kit, fill in the date and write down two things you learned that day.**

Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?



Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What did I learn today?

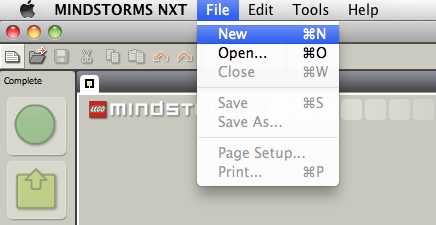


Today’s date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

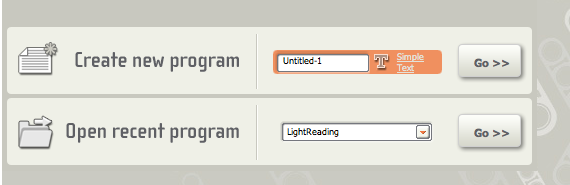
What did I learn today?



# How to Create a New Program

1. Go up to the top left of the screen and **click on “File”**
2. **Click on “New”**

**OR 1.** Begin typing a name for the new program in the box below.

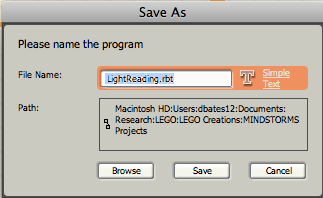


**OR 1.** Click on the “New File” Icon as show below

:New file_Icon.png

***New File*** 🡪

# How to Save Your Current Program



1. Go up to **“File”**
2. **Click on “Save”**
3. **Name your file and click “Browse”**
4. **Choose “Desktop”**
5. **Click “OK”**
6. **Click “Save”**

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your human robot.

### Goal

### Program the human robot to go across the room and sit down on a chair

### What functions will I use?

### Forward

Turn left 90 degrees

Turn right 90 degrees

Sit down

### Plan the process:

# Analysis

#### Use the following chart to record the changes you make. Once you get the robot to go across the room and sit on the chair can you make it more efficient?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it go across the room?** | **Did it sit down?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

### Program on the brick to make your robot

### Go from line a to line b

1. Go from line a to line b and return back to line a

### What functions will I use?

### Forward 5

Backward 5

Empty or Wait 2

Stop or Loop

### Plan the process:

# *Analysis*

#### Use the following chart to record the changes you make. Once you get the robot to go to line B and back can you make it more efficient?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it go from line a to line b?** | **Did it go from line b back to line a?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GOAL: Make a robot that can go from line A to line B, turn around, go back to line A, turn around, and stop.

# *Building*

1. Build the basic model from the Lego NXT Mindstorms 2.0 Education manual.

# *Functions*

New Functions that you’ll need for this robot:

### :::Desktop:Screen shot 2013-11-14 at 6.51.05 PM.png The Move Function

The move function controls the motors. It can make the robot move. This function determines which motors are running and with how much power and for how long.

Input: The signal to start

Process: Turns the motors on for the desired time

Output: The robot moves

### Move Function Variables

Port: Let’s you choose which motors to turn on

Direction: Choose which direction to go: forward, backward, or stop

Steering: Allows you to steer towards one motor

Power: Choose at how much power to run the motors

Duration: Choose how long to run the motor based on seconds, rotations, degrees, or run unlimited.

Next Action: Choose between brake and coast for the next action

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

### Make a robot that can go from line A to Line B, turn around, go back to line A, turn around, and stop

### What functions will I use?

### Move

### Plan the process:

# *Analysis*

#### Use the following chart to record the changes you make. Once you get the robot to go to line B and back can you make it more efficient?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it make it to line B?** | **Did it turn 180 degrees?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GOAL: Make a robot that moves forward when the touch sensor is pushed and stops when the sensor is released.

# *Building*

New pieces that you’ll need for this robot:

### :Touch Count Button:Single button_1.jpgTouch Sensor

The touch sensor can detect when the button is pushed down, released or pushed and released.

1. Plug the touch sensor into port 1.

# *Functions*

New Functions that you’ll need for this robot:

### The Loop Function

 The loop function causes all the functions that are placed inside it to repeat.

Input: The functions placed inside.

Process: Once it reaches the end of the loop, it goes back to the beginning of the loop.

Output: It repeats everything placed in the loop function.

### Loop Function Variables

Control: If this is set to forever, it will repeat forever. You can also set it to run a specific number of times or seconds.

### 

### The Switch function

The switch function works like the words EITHER…OR. It checks a *condition* and it EITHER does something if the condition is true, OR it does something else if the condition is false.

Input: Something from a sensor (light, sound, touch, ultrasonic)

Process: check to see if the sensor input matches the *conditions* you tell it.

Output: True or False

### Switch Function Variables

Control: Let’s you choose to wait for a sensor or wait for time

Sensor: Chooses which sensor to wait for

Display: Chooses whether to show the process for *true* and *false* conditions at the same time or separately.

Port: Indicates which port the sensor should be plugged into on the robot

Until: Indicates the *condition* to wait for. In this case, it’s waiting for light greater than 50%

Function: Tells the sensor to generate light or not.

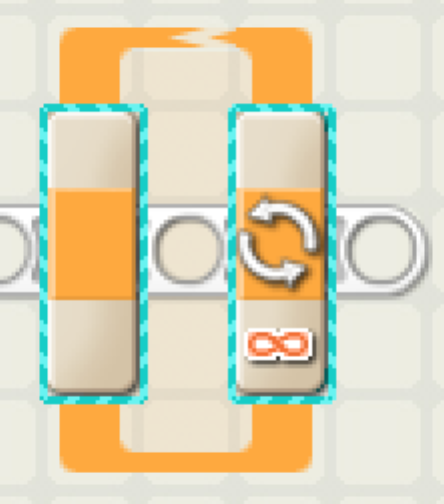
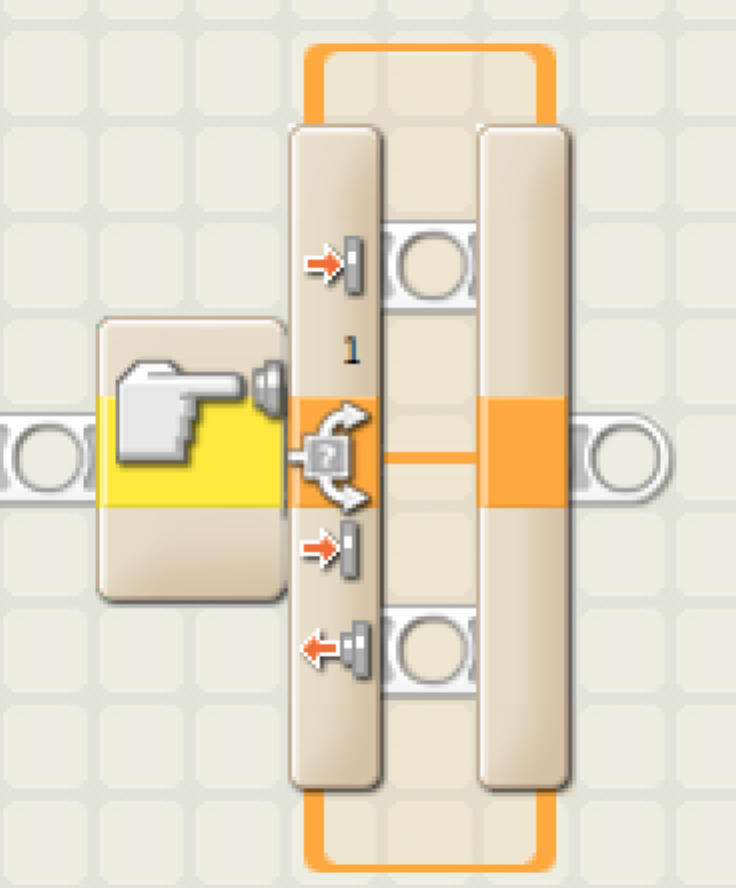
# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

### What sensors will I need?

### What functions will I use?

Move  Loop  Switch 

### Plan the process:

# *Programming*

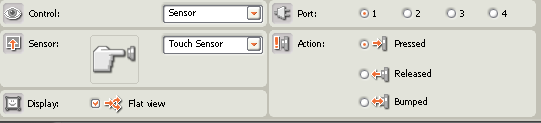
1. Place a **loop** on the canvas.

:::Desktop:Screen shot 2013-11-14 at 8.37.03 PM.png

1. Place a **switch** function inside the loop.



1. Set the **variables**.
   1. **Control:** Make sure this is set to sensor.
   2. **Sensor:** Make sure this is set to Touch Sensor**.**
   3. **Port:** Make sure port 1 is selected.
   4. **Action:** Make sure pressed is selected.

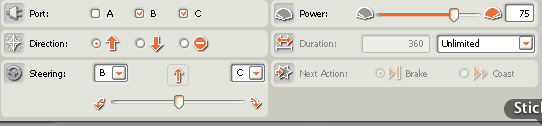




1. Place a **move** **function** inside the top section of the switch.

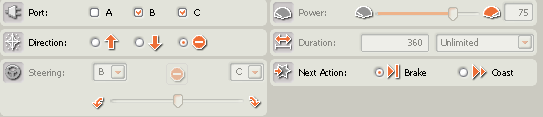


1. Set the **variables** to make it go forward.
   1. **Duration:** Set this to unlimited.
   2. **Port:** make sure these are set to the right ports



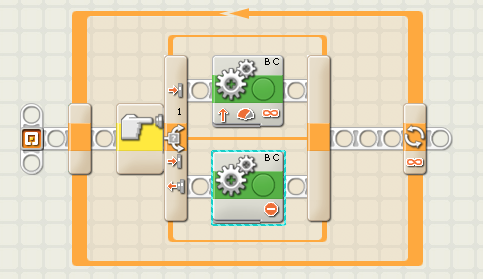
1. Place a **move** in the bottom section of the switch.

1. Set the **variables** so that it stops.
   1. **Direction:** Select the stop sign.



### Push and Go (final)

Your completed program should look like the following:



# *Analysis*

#### Use the following chart to record the changes you make. Once it works, what else can you do?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Does it go when you push down?** | **Does it stop when released?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GOAL: Create a robot that will follow a black line using the light sensor.

# Building the Line Following Robot

1. Complete pages 32-34 in your Lego Mindstorms Education manual.

### New pieces that you’ll need for this robot

##### Light Sensor

The light reading sensor detects how much reflected light there is on a surface. It provides a numeric value between 0% and 100%

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

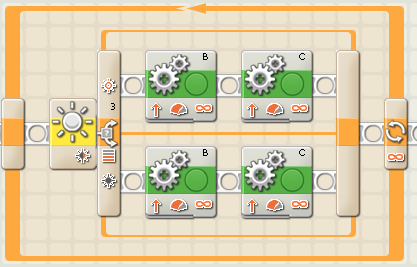
### What sensors will I need?

### What functions will I use?

### Plan the process:

# Programming

### Line Following Final



1. Follow the picture to set up the program
2. Set the variables of the **Switch** function
   1. Set **Sensor** to Light Sensor
   2. Set **Compare** to a number that is less than the reflected light from the table and more than the reflected light from the black line

*Hint: Use the brick to test the reflected light*

1. Set the variables for the **Move** functions
   1. Set **Duration** to unlimited for all move functions
   2. As seen in the picture select only one motor for each move function, so the first move function on top the **Port** should be set to B, and for the second one the **Port** should be set to C (repeat for the bottom Move functions.
   3. For the top move functions- set motor B **power** to greater than 50%, and motor C **power** to less than 50%
   4. For the bottom move functions- set motor B **power** to less than 50%, and motor C **power** to greater than 50%

Why are the **power** variables set to different numbers?

Can you make the robot complete the course faster?

# *Analysis*

Use the following chart to compare the times your robot gets on the different courses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | **Time** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GOAL: Create a robot that will start moving forward when it hears a sound and turn when it gets close to something, before continuing on.

# Building

1. Follow pages 24-26 of your Lego Mindstorms Education manual.
2. Follow pages 28-30 of your Lego Mindstorms Education manual.

New pieces that you’ll need for this robot:

##### Ultrasonic Sensor

The ultrasonic sensor detects how far away something is from the “eyes” of the sensor.

##### Sound Sensor

The Sound sensor will detect how loud a noise is.

# *Programming*

New Functions that you’ll need for this robot:

### The Switch function

The switch function works like the words EITHER…OR. It checks a *condition* and it EITHER does something if the condition is true, OR it does something else if the condition is false.

Input: Something from a sensor (light, sound, touch, ultrasonic)

Process: check to see if the sensor input matches the *conditions* you tell it.

Output: True or False

### Switch Function Variables

Control: Let’s you choose to wait for a sensor or wait for time

Sensor: Chooses which sensor to wait for

Display: Chooses whether to show the

Port: Indicates which port the sensor should be plugged into on the robot

Until: Indicates the *condition* to wait for. In this case, it’s waiting for light greater than 50%

Function: Tells the sensor to generate light or not.

# *Functions*

New Functions that you’ll need for this robot:

### The WAIT function

The Wait function is used to tell your robot to wait for a certain input before proceeding. You can choose to have the robot wait for a certain period of time or to wait for any sensor.

Input: Something from a sensor (light, sound, touch, ultrasonic)

Process: check to see if the sensor input matches the conditions you tell it.

Output: Advance the robot to the next function

### Wait Function Variables

Control: Let’s you choose to wait for a sensor or wait for time

Sensor: Chooses which sensor to wait for

Port: Indicates which port the sensor should be plugged into on the robot

Until: Indicates the *condition* to wait for. In this case, it’s waiting for light greater than 50%

Function: Tells the sensor to generate light or not.

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

### What sensors will I need?

### What functions will I use?

### Plan the process:

# *Analysis*

#### Use the following chart to record the changes you make. Once it works, what else can you do?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it respond to a sound by moving?** | **Did it turn when it got close to something?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GOAL: Make a robot that reads and displays reflected light.

# *Building*

1. Attach the light sensor.

# *Functions*

New Functions that you’ll need for this robot:

### :::Desktop:Screen shot 2013-12-03 at 12.57.19 PM.pngThe DISPLAY function

The Display function allows something to be displayed on the screen on the brick.

Input: A variable

Process: communicates with the brick

Output: Displays something on the screen

### Display Function Variables

Action: Let’s you choose what type of information will be displayed

Display: Determines if the screen should be cleared first

File/Text/Type: Set what is displayed

Position: Set the position on the screen where something is displayed

### :::Desktop:Screen shot 2013-12-03 at 1.15.54 PM.pngThe NUMBER TO TEXT function

The Number to Text function is used change a number variable into a text variable.

Input: A number variable

Process: Converts a number variable to a text variable

Output: A text variable

### Number to Text Function Variables

Number: Let’s you manually set the number

### :::Desktop:Screen shot 2013-12-03 at 1.30.07 PM.pngThe LIGHT SENSOR function

The Light Sensor function uses the Light Sensor to detect how much light is reflected.

Input: Light from the light sensor

Process: Converts the amount of reflected light into a number

Output: A number variable

### Light Sensor Function Variables

Port: Let’s you choose which port the light sensor is in

Compare: Set what is considered light or dark

Function: Choose whether or not the light sensor generates light

# *Planning*

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal

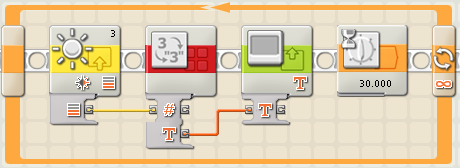
### What sensors will I need?

### What functions will I use?

### Plan the process:

# Programming

### Light Reading Final



1. Set your program up like the picture.
2. Connect the wires
3. Set the variables for the **Light** function
   1. **Port:** Set to which port you are using.
   2. **Compare:** Leave default settings**.**
   3. **Function:** Select Generate Light.
4. Set the variables for the **Display** function
   1. **Action:** Set to Text



1. Set the variables for the **Wait** function
   1. **Control:** Set to time
   2. **Until:** Set how much time you want it to wait before continuing on

# RECORDING THE LIGHT READING

# *Analysis*

Instructions: For this activity, you will explore how much light comes from the different colors of the different objects in the room. Use the table below to record the value of different colored items throughout the room. Try to measure a range of light to dark objects.

|  |  |  |
| --- | --- | --- |
| **Location** | **Color** | **Light reading** |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |
| 10. |  |  |

What could this be used for in real life?



How could this program be improved?



GOAL: Create a robot that will count how many times a person can press a button in 10 seconds

# *Building*

1. Attach the touch sensor.

# *Functions*

New Functions that you’ll need for this robot:

### :::Desktop:Screen shot 2013-12-03 at 12.57.19 PM.pngThe SOUND function

The Sound function allows a tone or sound file to be played.

Input: A variable

Process: communicates with the brick

Output: Displays something on the screen

### Display Function Variables

Action: Let’s you choose what type of information will be played

Control: Determines if the sound starts or stops playing

Volume: Adjusts the volume

Function: Allows it to be set to repeat

File: Choose sound file

Wait: Allows the program to wait until the sound is done playing

# Planning the Touch Count Robot

The first part in creating any program is to plan it out. Use this sheet to plan out your robot.

### Goal?

### What sensors will I need?

### What functions will I use?

### Plan the process:

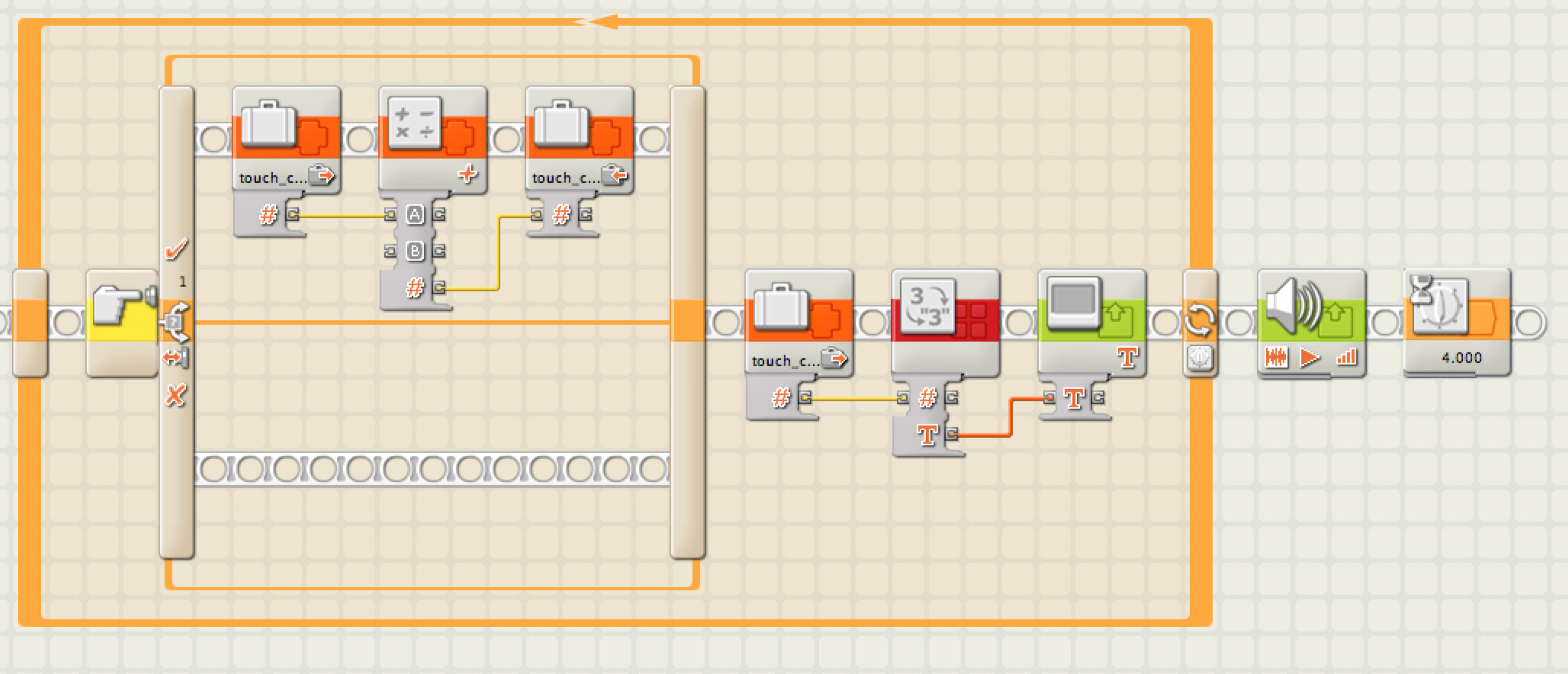
# *Analysis*

#### Use the following chart to record the changes you make. Once it works, what else can you do to make it more efficient? Explore other sensors or make up your own variable.

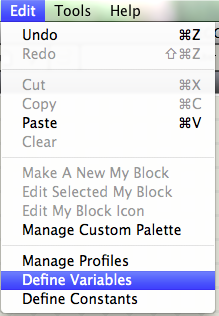
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it count the number of presses for the button?** | **Did it display a count and stop after 10 seconds?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### Touch Count (final)

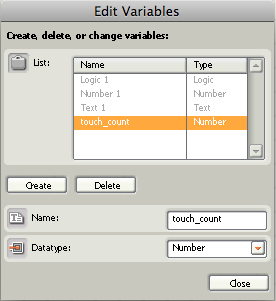
Goal: Create a robot that will count how many times a person can press a button in 10 seconds



1. Set up the program in the picture
2. Create a new variable
   1. Go to **Edit** and select **Define Variable.**



* 1. Click the **Create** button.
  2. Name the variable **touch\_count**.
  3. Set the **Datatype** to **Number**.
  4. Click **Close**.



1. Set the **Loop** to repeat for 10s.
2. **Switch** function
   1. Set Action to **Bumped**
3. All **Variable** functions
   1. Select **touch\_count**
4. 2nd **Variable** function
   1. Set Action to **Write**
5. Connect the wires
6. **Math** function
   1. Set B to **1**
7. **Display** function
   1. Set Action to **Text**
8. Choose a sound file for the **Sound** function.
9. Set the **Wait** function to 4s.

How many times can you push the button in 10s?

*Challenge*

1. Have it add 2 points for every click.
2. Use a different sensor and create an appropriate variable.

GOAL: Create a robot that will go back and forth between two pieces of black tape as many times as possible in 15 seconds.

# *Building*

1. Attach the light sensor.

# *Functions*

# Planning the Back and Forth robot

### Goal?

### What sensors will I need?

### What functions will I use?

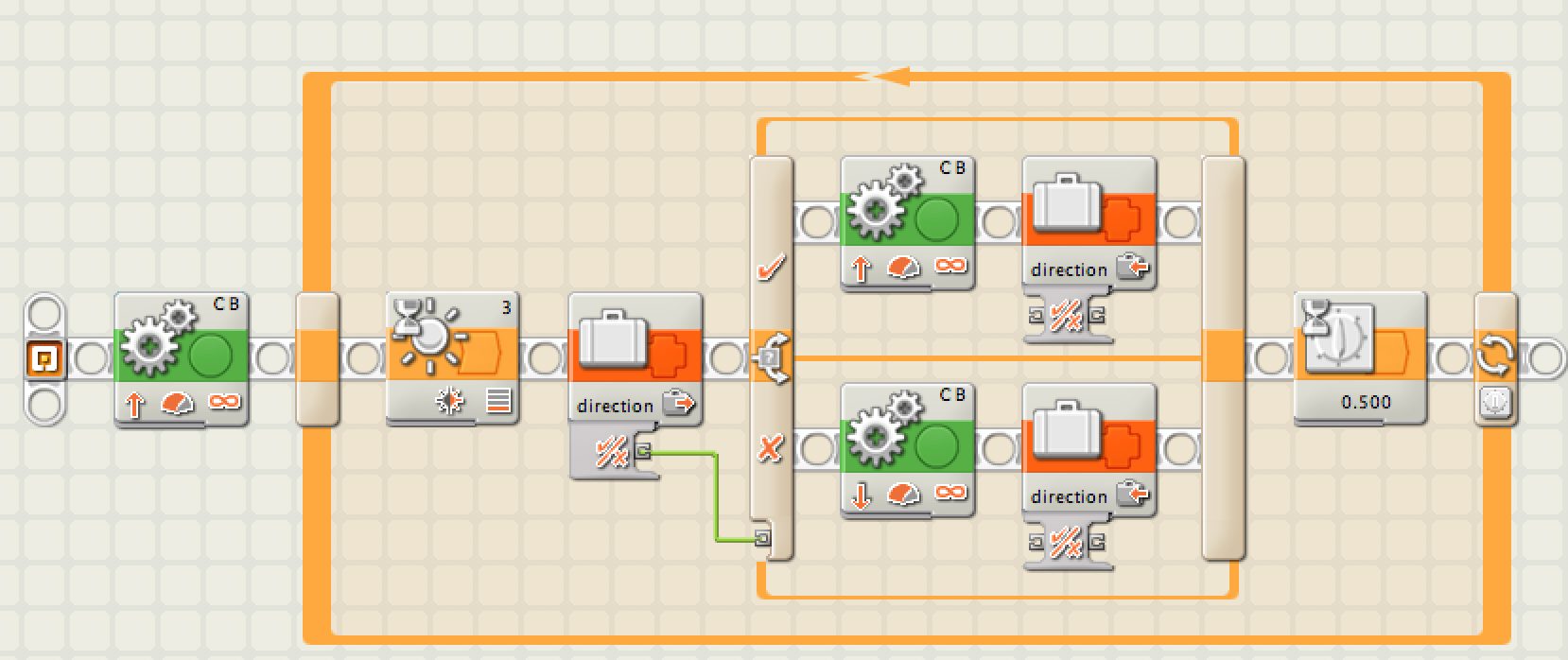
### Plan the process:

# *Analysis*

#### Use the following chart to record the changes you make. Once it works, what else can you do to make it more efficient? Explore other sensors to use. Is there a way to count how many times it went back and forth?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Did it go back and forth between two black pieces of tape?** | **Did it loop back and forth until the end of 15 seconds?** | **Changes made** | **Reason** | **Result** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### Complete Back and Forth program

Your completed Back and Forth program should look something like the following. Download it to your robot and try it out!

CHALLENGE In an earlier program (touch count), we made it so the robot could count how many times it did something and we displayed that on the screen. Can you make it so the Back and Forth robot shows you how many times it changes direction?