

Building a LEGO Light/Color sensor Trigger Assembly And NXT/EV3 programming

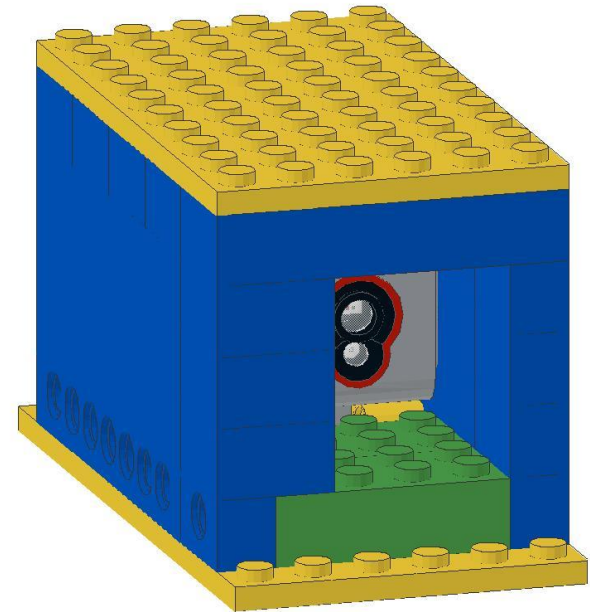
By: Tom Bickford
Maine Robotics

www.mainerobotics.org

© 2017

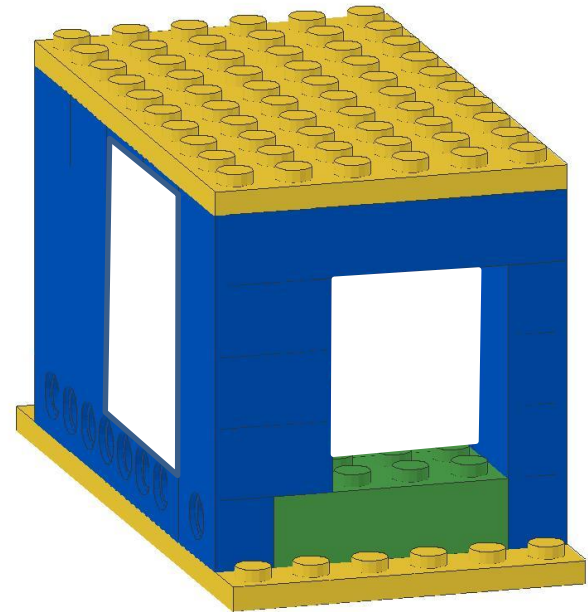
Making a LEGO® Light/Color Sensor Trigger

- This design uses a LEGO light sensor and a light source to provide a trigger to the NXT or EV3 controller.
- Two sources of light can be used:
 - LASER* line level gives a very definitive line or dot to use as the light source
 - High intensity flashlight
- Note: If you use the larger plate on the bottom, it gives you something to tape down when you need to secure this to a surface.
- * LASERs (light amplification by stimulated emission of radiation) are dangerous to the eye. Proper precaution (including LASER rated safety glasses, and adult supervision) are necessary to maintain a safe environment.



Conditioning the light

- When using the light sensor, if you build the sensor box to allow for the suspension of a piece of white paper, approximately 1 or 2 bricks from the sensor front, it will allow more light to be seen by the sensor.
- Without this diffuser, if the light source is slightly off center, it will miss being read by the sensor.
- We use standard copy paper, cut to size inserted between the bricks and folded back and taped down.



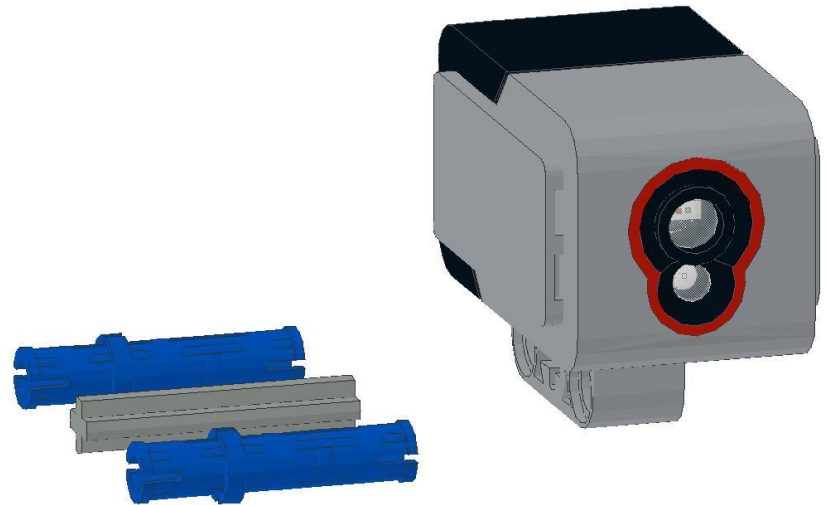
Step 1

- Start with your sensor.
- Depending on the sensor or the year it was made, will depend on its size
- This example is using an EV3 light sensor



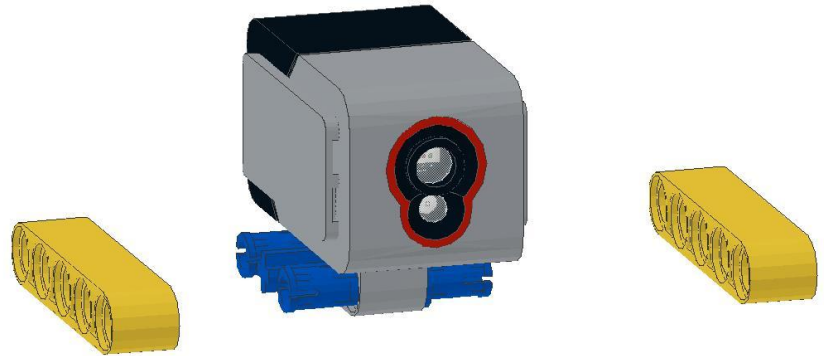
Step 2

- Fill the three holes in the sensor with long pins or long pins and #3 technic axles



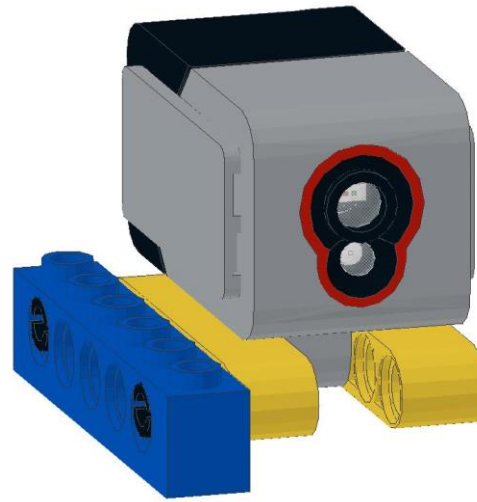
Step 3

- Add two #5 technic lift arms to the sensor base, using the 3 middle holes of the lift arms



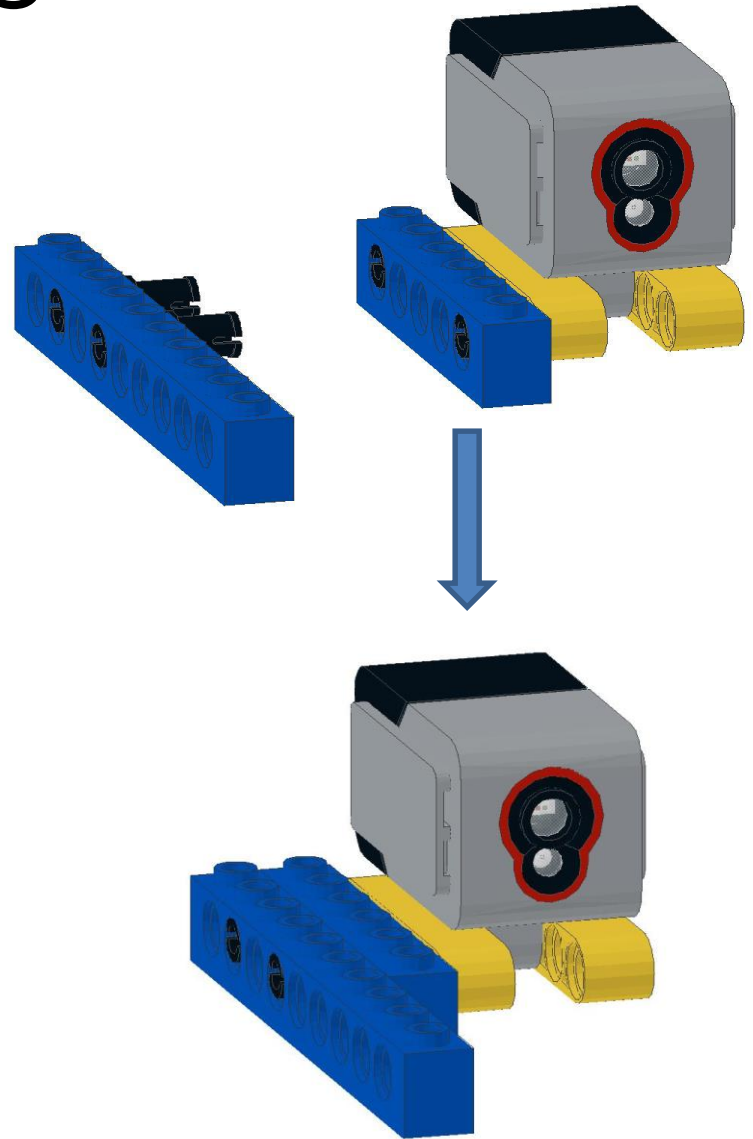
Step 4

- Add a 1x6 technic brick to one side of the sensor assembly using two short friction pins.



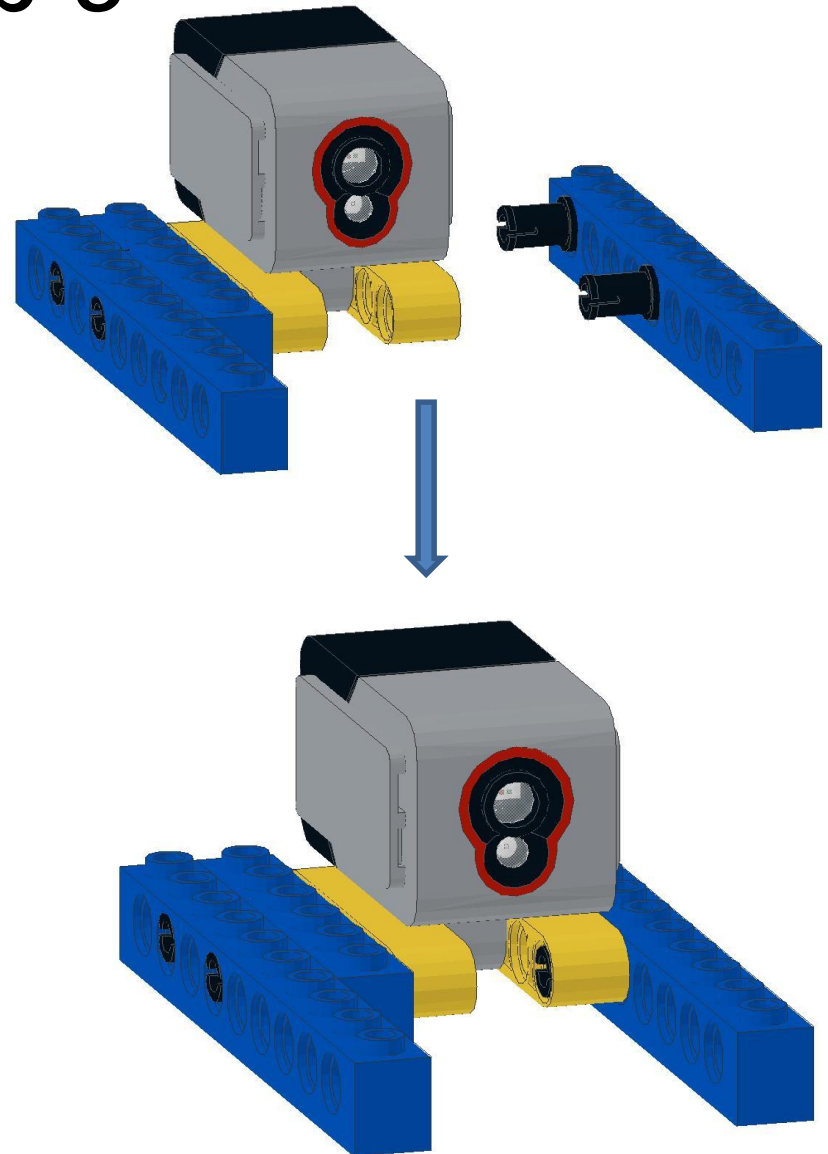
Step 5

- Add a 2x10 technic bricks using 2 short friction pins in holes #2 and #4



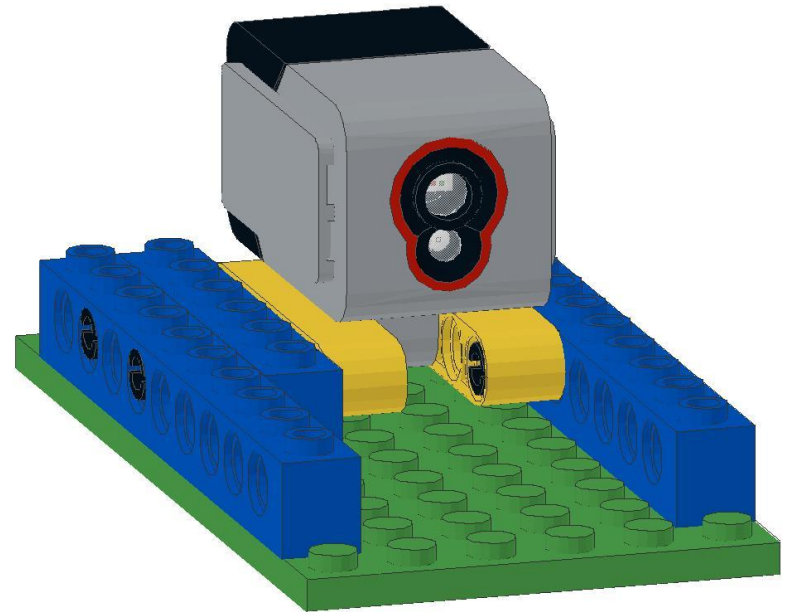
Step 6

- Add a second 1x10 technic brick using two short friction pins in holes 1 and 5



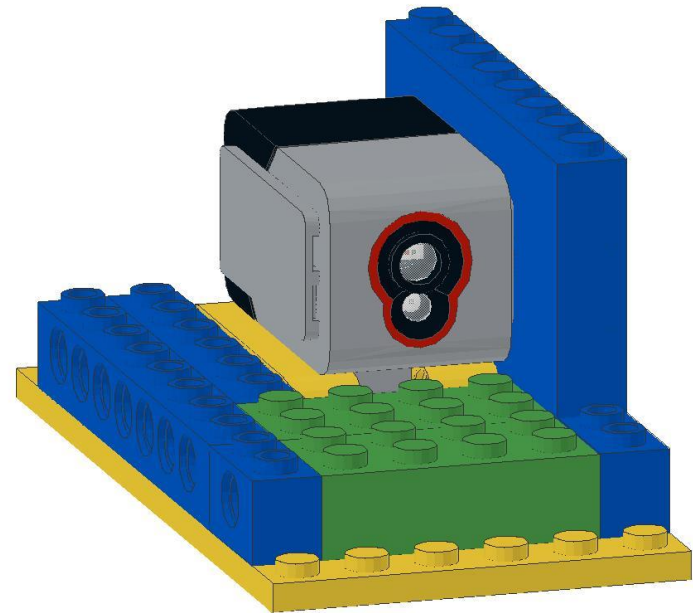
Step 7

- Add a 6x12 plate to the bottom of the assembly, with one pin sticking out the front and back



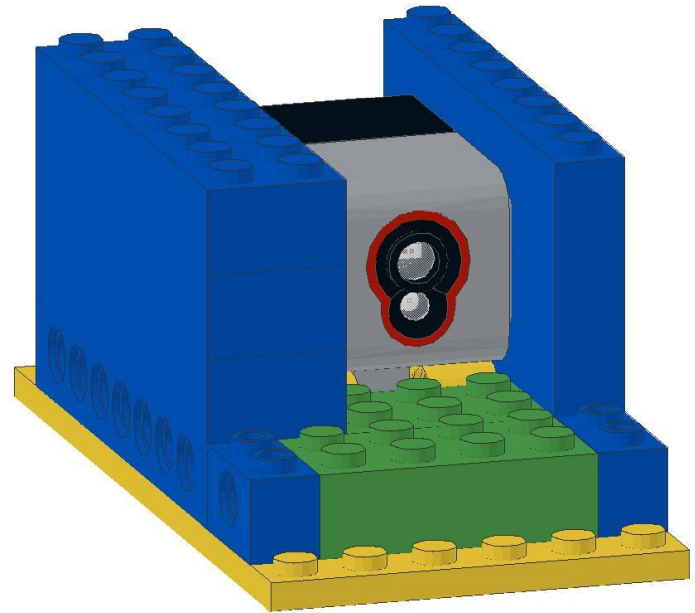
Step 8

- Add three 1x8 bricks. You can use either whole bricks or technic bricks, but whole bricks are better, they don't let the light in through the sides.
- Add two 2x4 bricks to base



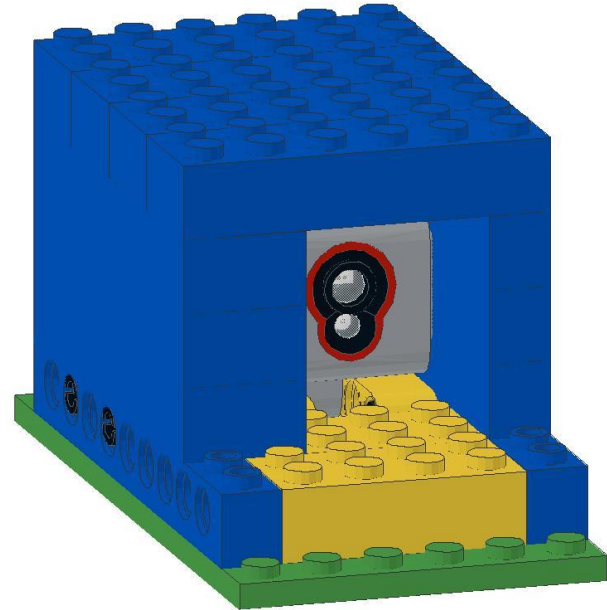
Step 9

- Add three 2x8 bricks to the side of the assembly



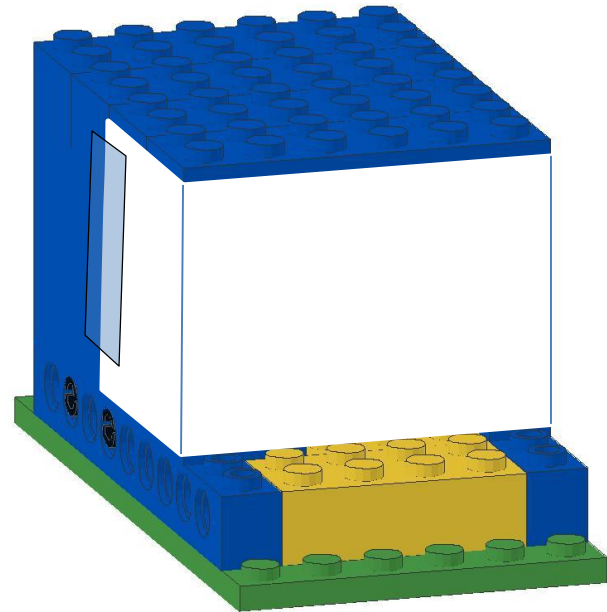
Step 10

- Place four 2x6 bricks across the top of the assembly from side to side



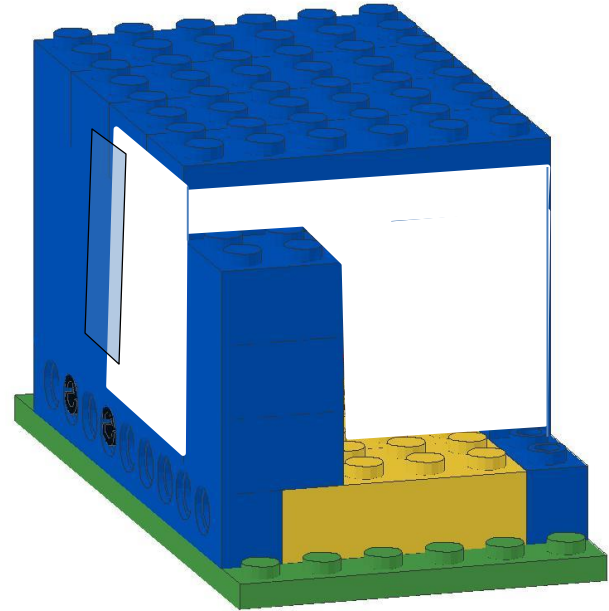
Step 11

- Cover the opening to the front of the sensor with a piece of thin white paper.
- Bend it back out of the way and tape it down.



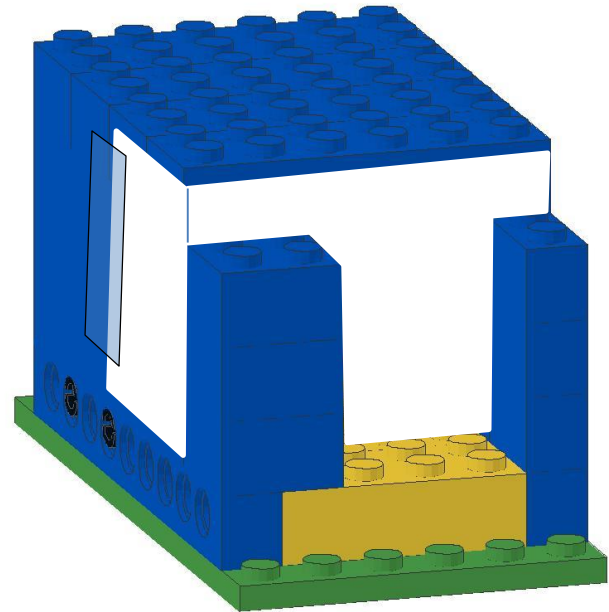
Step 12

- Add three 2x2 bricks



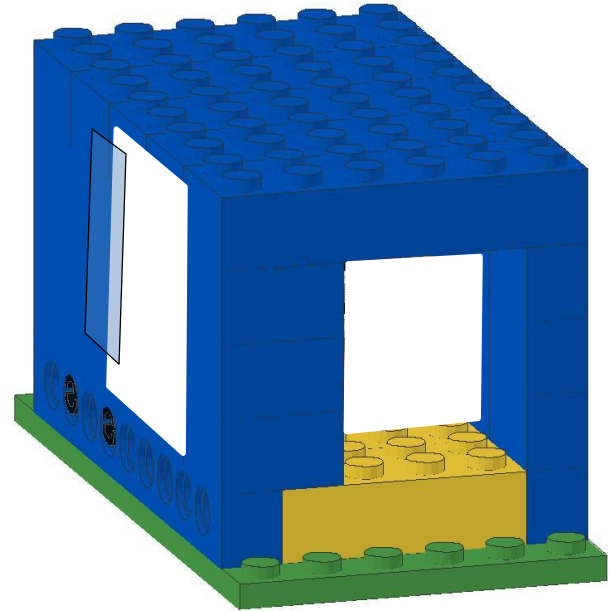
Step 13

- Add three 1x2 bricks



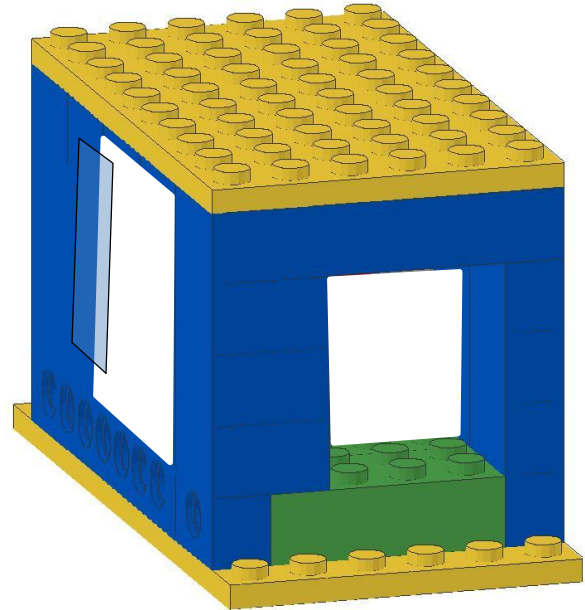
Step 13

- Add a single 2x6 brick to complete the top



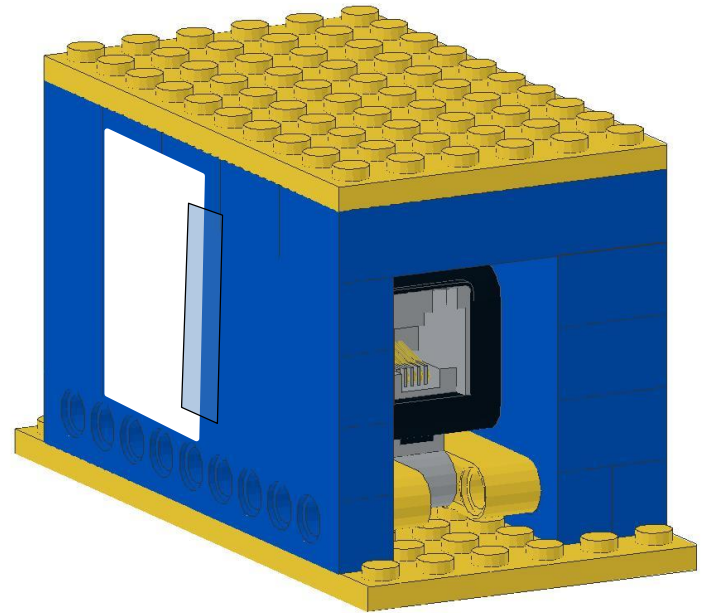
Step 14

- Place a 6x10 plate on the top of the assembly



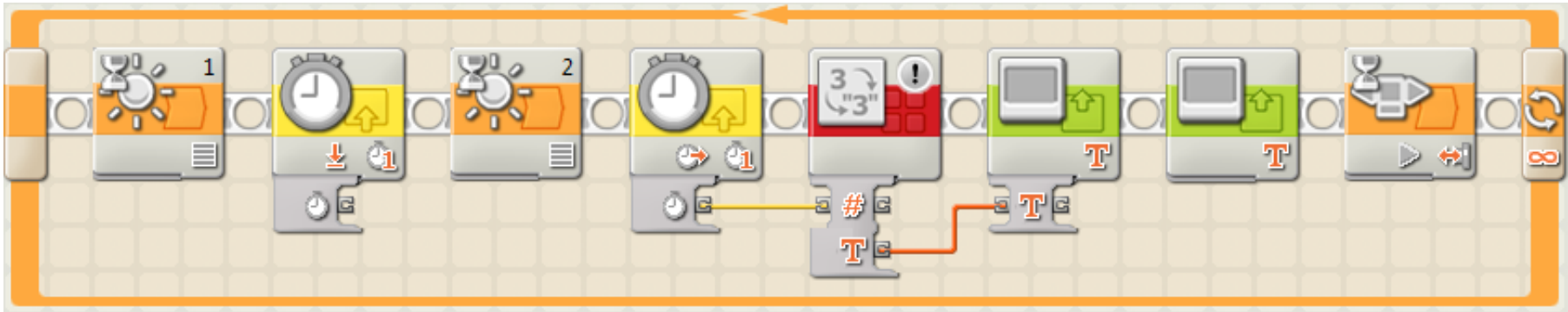
Step 15

- Flip around and attach your wire back to the controller



NXT Race Timer Program

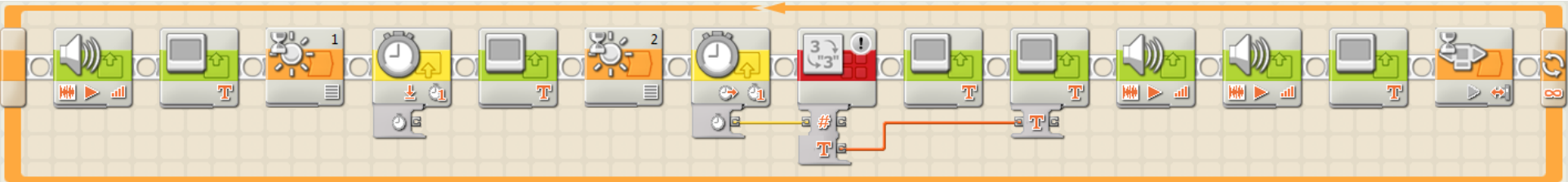
Minimal



- In this loop, the program uses two sensors (light triggers) to start and stop the #1 timer
 - Wait for light sensor 1 to go below a value (set to appropriate value to be true when you cross the window of the sensor)
 - Reset the timer
 - Wait for light sensor 2 to go below a value
 - Read the timer
 - Convert the timer number to text
 - Show the text on the display screen (clear screen)
 - Display a continue option “Press > to continue” (don’t clear screen)
 - Wait for the right button of the NXT to be pressed to reset the system
 - Note: unless cleared, the system will keep showing the last screen until you override it with the next display command.

NXT Race Timer Program

Full program

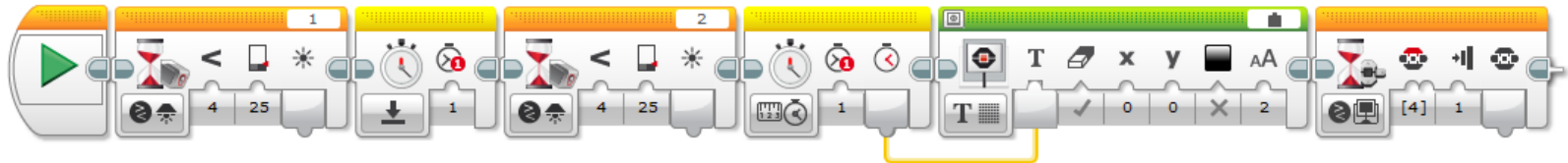


So when you start the program, you enter a loop. A number of display screens tell you what to do, along with audio prompts

- First thing it does is say "Start" then shows "Ready to go..." on the screen <clear display checked>.
- It then waits for a low value at the first gate (the start line) caused by something (the robot) crossing in front of it.
- Once triggered it resets a timer, in this case Timer #1
- Then it displays "Running..." <clear display checked> and
- It then waits for a low value at the second gate (the finish line) caused by the robot crossing in front of it.
- Once the second light sensor is triggered:
 - The timer value is read
 - A display block writes "Time (milliseconds)" at the top of the screen <clear display checked>
 - This number (in milliseconds) is given to the "number to text" block
 - The text value is then given to the display block <clear display unchecked>
 - It then says "Stop" and "Score" and displays "Press > to continue" <clear display unchecked>
- And then waits for an NXT right arrow push before going back to the beginning.
- Pay close attention to which display blocks have a clear display checked and which don't.
- Also you'll have to play with light levels on the two light sensors.

EV3 Race Timer Program

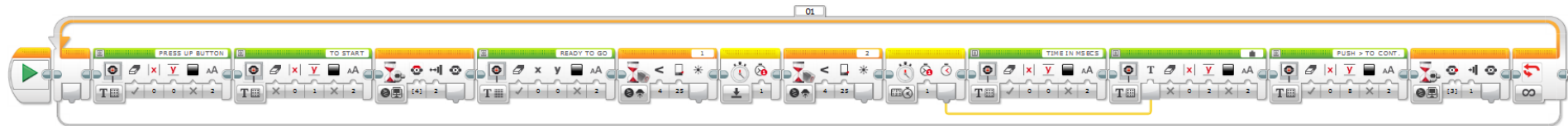
Minimal



- In this program two sensors (light triggers) start and stop the #1 timer
 - Wait for light sensor 1 to go below a value (set to appropriate value to be true when you cross the window of the sensor)
 - Reset the timer
 - Wait for light sensor 2 to go below a value
 - Read the timer
 - Show the time on the display screen (clear screen)
 - Select “Wired” in the upper right corner box on the display button
 - Connect the output from Timer1 to the Text input on the display button
 - Wait for the top button of the EV3 to be pressed to reset the system

EV3 Race Timer Program

Full program



So when you start the program, you enter a loop. A number of display screens tell you what to do

- First thing it shows "PRESS UP BUTTON TO START" on the screen <clear display checked>.
- Then it waits for the up button to be pressed and released
- It then waits for a low value at the first gate (the start line) caused by something (the robot) crossing in front of it.
- Once triggered it resets a timer, in this case Timer #1
- Then it displays "Running..." <clear display checked> and
- It then waits for a low value at the second gate (the finish line) caused by the robot crossing in front of it.
- Once the second light sensor is triggered:
 - The timer value is read
 - A display block writes "time in msec's" at the top of the screen <clear display checked, y=0>
 - This number (in milliseconds) is written to a lower line <clear display unchecked, y=2>
 - It then displays "Press > to continue" <clear display unchecked, y=6>
- And then waits for an EV3 right arrow push before going back to the beginning.
- Pay close attention to which display blocks have a clear display checked and which don't.
- Also you'll have to play with light levels on the two light sensors.
- You can also add sounds, including "start", "stop", "ready" or other cues to help you navigate
- This program allows you to keep running trials without restarting the program