Table of contents

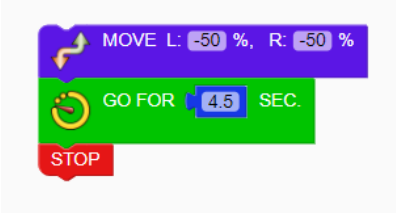
Sample Code - 2D robots

Move Forward	
Blocks	javaScript

	//Set motor power
	move(50, 50);
	
	//continue for 3 seconds
	await gofor(3);
	
	// Reset power to zero
	stop();

Left and Right motors have equal, and positive power	

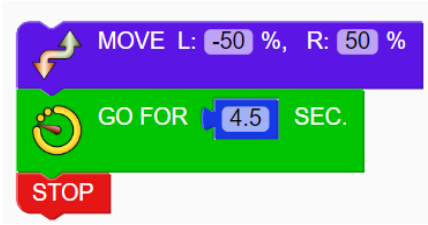
Move Backward	
Blocks	javaScript

	//Set motor power
	move(-50, -50);
	
	//continue for 3 seconds
	await gofor(3);
	
	// Reset power to zero
	stop();

Left and Right motors have equal, and negative power	

Left Turn

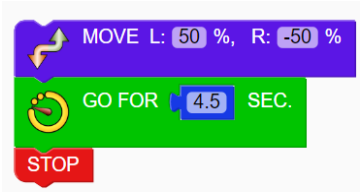
Blocks	javaScript

	//Set motor power
	move(-50, 50);
	
	//continue for 3 seconds
	await gofor(3);
	
	// Reset power to zero
	stop();

Differential power (left power - right power) is negative	

Right Turn

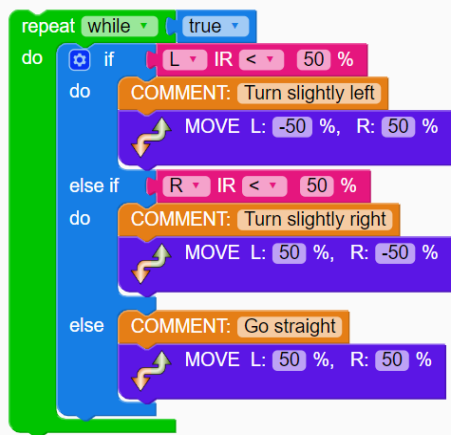
Blocks	javaScript

	//Set motor power
	move(50, -50);
	
	//continue for 3 seconds
	await gofor(3);
	
	// Reset power to zero
	stop();

Differential (left - right) power is positive	

Following a dark line

Blocks



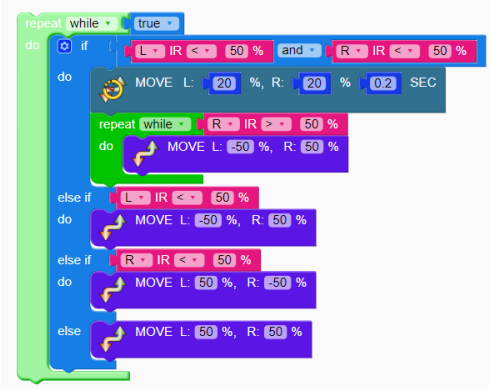
javaScript

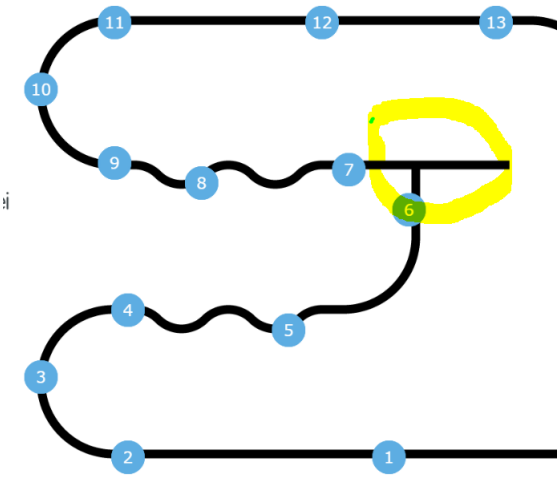
```
while (true)
{

// Must read inputs in the loop for latest status
// readInputs() also needs await keyword

    await readInputs();

    if( irLeft() < 50)
    {
        move(-50, 50);
    }

    else if ( irRight() < 50)
    {
        move(50, -50);
    }

    else
    {
        move(50, 50);
    }
}
```

Adjust the power to move faster.
Adjust the differential power to get sharper turns

Navigating the T-junction

Blocks





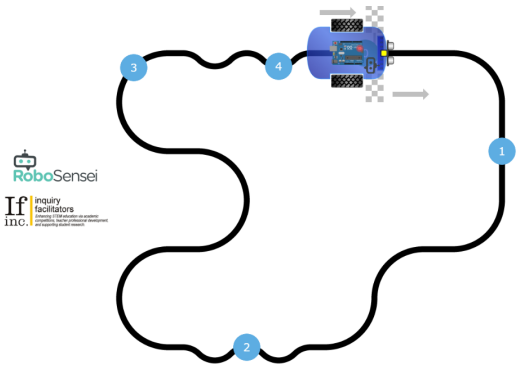
javaScript

```
while (true)
{
  // Must read inputs in the loop to get the latest status
  // readInputs() also needs await keyword
   await readInputs();

   // If BOTH sensors on the line, it is a T-junction
   if ( ( irLeft() < 50 ) && ( irRight() < 50 ) )
   {
      TLeft();
   }

   // Check if left sensor is on or near the line
   else if ( irLeft() < 50 )
   {
      move(-40, 70);
   }

   // Check if right sensor is near the line
   else if (irRight() < 50)
   {
      move(70, -40);
   }

   // Otherwise keep going straight
   else
   {
      move(100, 100);
   }
}

// Function to handle T-junction followied
// by line seek to the left
async function   TLeft()
{
    move(20, 20);
    await gofor(.2);

     while( irRight() <  50 )
     {
       move(-40, 40);
       // We must get the latest input state
       // before the next loop iteration
       await readInputs();
     }
}
```

Adjust the power to move faster and get sharper turns.

Line following with Sensor Array and PID control algorithm

Blocks	javaScript

	// PID Control example with RoboSensei IR sensor array
	
	var kp = 5;
	var ki = 5;
	var kd = 3;
	
	var P = 0;
	var I = 0;
	var D = 0;
	
	var prevError = 0;
	var correction = 0;
	
	var lp = 70; // left power
	var rp = 70; // right power
	
	var error = 0;
	
	while (true)
	{
	  // Must read inputs in the loop to get the latest status
	  // readInputs() also needs await keyword
	  await readInputs();
	
	  error = irArrayErrorGet();
	
	  if (error == 255)
	  {
	  tprint('oops! out of line!')
	  }
	
	  P = error;
	  I = I + error;
	  D = prevError - error;
	  prevError = error;
	
	  correction = kp * P + ki * I + kd * D;
	  move(lp - correction, rp + correction);
	}

Google / checkout wikipedia to learn more about PID control algorithms. Your air conditioner will be using this algorithm.
Experiment with tuning kp, ki, and kd to improve the line tracking performance. Start with slow speed, slowly increase the speed and tune the control loop.

Sample Code - 3D robots

First candle - find, drive to and extinguish

```
await turn_to_flame(-20);
await Sleep(1);
await go_to_line(20);
await Sleep(1);
await extinguish();
await Sleep(1);

//_____________FUNCTIONS_____________

async function turn_to_flame(v)
{
    while(Robot.FlameSensor < 50)
    {
        await Robot.ReadInputs();
        Robot.SetPower(-v, v);
    }
    Robot.SetPower(0, 0);
}

async function go_to_line(v)
{
    while(Robot.IrLeft > 10)
    {
        await Robot.ReadInputs();
        Robot.SetPower(v, v);
    }
    Robot.SetPower(0, 0);
}
```

```
async function extinguish()
{
    Robot.FanOn();
    await Sleep(2);
    Robot.FanOff();
}
```

JavaScript Guide - 2D robots

RoboSensei JavaScript functions

API	Use notes
await readInputs()	usage: await readInputs(); This function needs to be called before using sensors in the code. In case of loops, where sensor is used; call this at the beginning, inside the loop.
move(powerLeft, powerRight)	usage : move(powerLeft, powerRight); powerLeft: left motor power 0 - 100 powerRight: right motor power 0 - 100
await movefor(powerLeft, powerRight, timeSeconds)	usage: await movefor(powerLeft, powerRight, timeSeconds); powerLeft: left motor power 0 - 100 powerRight: right motor power 0 - 100 timeSeconds: time in seconds (decimal number) Example: await movefor(20, 20, 2);
await delay(timeSeconds) await gofor(timeSeconds)	Usage: await delay(timeSeconds); Usage: await gofor(timeSeconds); timeSeconds: time in seconds (decimal number)
stop()	Usage: stop() Stops the robot by applying zero power to left and right motors. Same as move(0,0);
irLeft()	Usage: irLeft(); Returns the value of the LEFT IR sensor

irRight()	Usage: irRight() Returns the value of the RIGHT IR sensor
distance()	Usage: distance(); Returns value of distance sensor in mm.
bumperLeft() bumperRight() **Xplorer ONLY**	Usage: bumperLeft(); bumperRight() Returns **true** : bumper is hit; **false**: bumper not hit

ledon()
ledoff()
 	
Xplorer ONLY
	Usage:
	
	ledon(ledId); ---> ledon(1);
	ledoff(ledId; ----> lefoff(1);
	
	Red: 0, Yellow: 1, Green: 2, Left: 3, Right: 5

tprint(text)	Usage: tprint('text'); Prints text to the debug console located at the bottom of the the javascript window You can also print sensor values. Example: tprint('Distance: ' + distance()); will print distance sensor value Caution: ***tprint*** command severely impacts the robot performance, hence should not be used in loops where high performance sensor reading and code execution is required for example in a line follower loop. Generally only use this for debugging purposes.

rgbleft()
rgbright()
 	
mBot ONLY
	Usage: rgbleft(rValue, gValue, bValue); rgbright(rValue, gValue, bValue);
	Controls the RGB LEDs on mBOT.
	
	rValue, gValue, bValue: 0 to 255
	
	Sample code to blink left RGB LED:
	
	while(true)
	{
	rgbleft(255, 0, 0);
	await delay(1);
	rgbleft(0,0,0);
	await delay(1);
	}

JS API for the IR sensor array.

<table>
<tr><th>API</th><th>Use notes</th></tr>
<tr><td>ir8Array()</td><td>var sensorArray = ir8Array();

Returns an array of 8 integers whose value is between 0 (black color) and 255(white color)</td></tr>
<tr><td>ir8Byte()</td><td>var sensorByte = ir8Byte();

Returns an integer, whose least significant byte represents a 1/0 status of each sensor based on threshold value of 100.

<table>
<tr><th colspan="4">Left half</th><th colspan="4">Right half</th><th></th></tr>
<tr><th>S7</th><th>S6</th><th>S5</th><th>S4</th><th>S3</th><th>S2</th><th>S1</th><th>S0</th><th></th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>Reading >= 100</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Reading < 100</td></tr>
</table>
</td></tr>
<tr><td>irArrayErrorGet()</td><td>var error = irArrayErrorGet()

This function will return an error value based on the detected sensor values.

This is a convenience function. You may want to write your own function.
This function return the following values:

In the following table ‘x’ (don’t care) means value does not impact the returned value. This could be a 0 or 1.

<table>
<tr><th>S7</th><th>S6</th><th>S5</th><th>S4</th><th>S3</th><th>S2</th><th>S1</th><th>S0</th><th>Returned error value value</th></tr>
<tr><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>255</td></tr>
<tr><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>4</td></tr>
<tr><td>x</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>3</td></tr>
<tr><td>x</td><td>x</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>2</td></tr>
</table>
</td></tr>
</table>

	x	x	x	0	x	x	x	x	1
	x	x	x	x	0	x	x	x	-1
	x	x	x	x	x	0	x	x	-2
	x	x	x	x	x	x	0	x	-3
	x	x	x	x	x	x	x	0	-4

Advanced: Using JavaScript Timers

```
// ASYNCHRONOUS PROGRAMMING WITH JS TIMERS
// To see the following example, use xplorer robot to see red light blink with 500 ms interval

var myVar = setInterval(myTimer, 500)

// We need to make sure the program does not exit
while (true)
{
        await delay(3);
}

// Function to run on the timer interval
var isOn = false;
function myTimer()
{
   if(isOn)
   {
      ledoff(0);
   }
   else
   {
     ledon(0);
   }

  isOn = !isOn;
}
```

JavaScript Guide - 3D robots

Driving

```
Robot.SetPower(left, right)     // power on each wheel from -100 to 100
```

Waiting

```
Sleep(seconds)                          // need await before it
```

Fan

```
Robot.FanOn()                   // switching the fan
Robot.FanOff()
```

Sensors

```
Robot.ReadInputs()              //sensor values actualization
Robot.FlameSensor               // range 0 - 100
Robot.IrLeft                    // range 0 - 255
Robot.IrRight                   // range 0 - 255
Robot.Distance                  // measured in mm
```