LOOKING FOR A OBJECT

STEMJAM Teaching Guide

Developing make spaces to promote creativity around STEM in schools
Acronym: STEMJAM
Project no. 2016-1-ES01-KA201-025470

www.stemjam.eu

Co-funded by the Erasmus+ Programme of the European Union

LOOKING FOR A OBJECT

ABSTRACT

The idea is to program mBot to search a limited area

Using a robot to find a lost object in limited area. This area is limited by black line. The object could be a high block or bottle and it is found using ultrasonic sensor. Use compass to keep direction.

In a second version, the area is not limited in terms of “physical” borders as black lines, but his extension is given by time.

We used compass to keep direction and using angles to change it.

DIDACTIC OBJECTIVES

Students learn about how to:

- Using compass.
- The repeat forever – loop condition.
- Understand angles.

STEM Subject: Science☐ Technology ☒ Engineering☐ Mathematics☐

Education Level: 12-14 years☒ 14-16 years☒

PROBLEM STATEMENT

First version

Using a robot to find a lost object in limited area. This area is limited by black line. The object could be a high block or bottle and it is found using ultrasonic sensor. Use compass to keep direction.

Second version

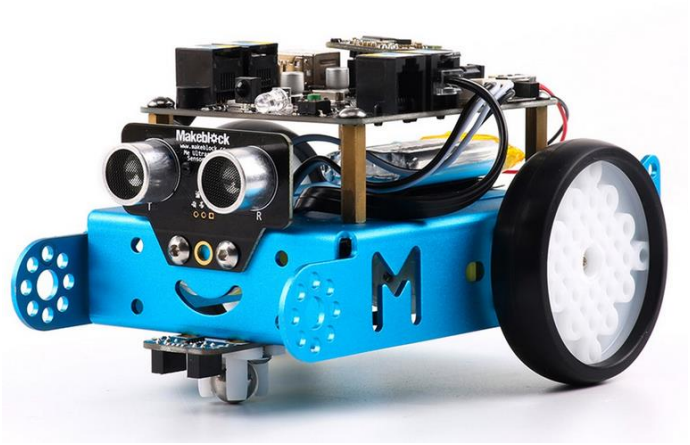
Using a robot to find a lost object in limited area. The area is not limited in terms of “physical” borders as black lines, but his extension is given by time.

We used compass to keep direction and using angles to change it.

BOM (Bill of Materials Needed)

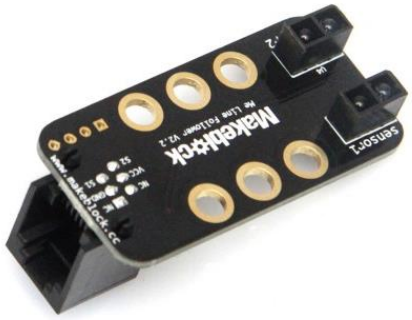
The electronic components and devices needed for this activity are:

- mBot => Ref. 90054



- Me **Ultrasonic Sensor**:

- Me **Line Follower**:



- Board with limited area.
- Compass.

First version

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT 1	P.MOT 2
				Y	B	W	Y	B	W	Y	B	W	BI	Y	B	W	BI	W*	W*
Mbot Robot 2´4G																			
Motor 1	W*		1															W*	
Motor 2	W*		1																W*
Me 7-Segment serial display	B		1		B														
Me Ultrasonic sensor	Y		1							Y									
Me Line Follower	B		1					B											
Me Compass	W		1												W				
RJ25 cables			4																
Structures and beams																			
Laptops			1																
Attrezzo (not essential)																			

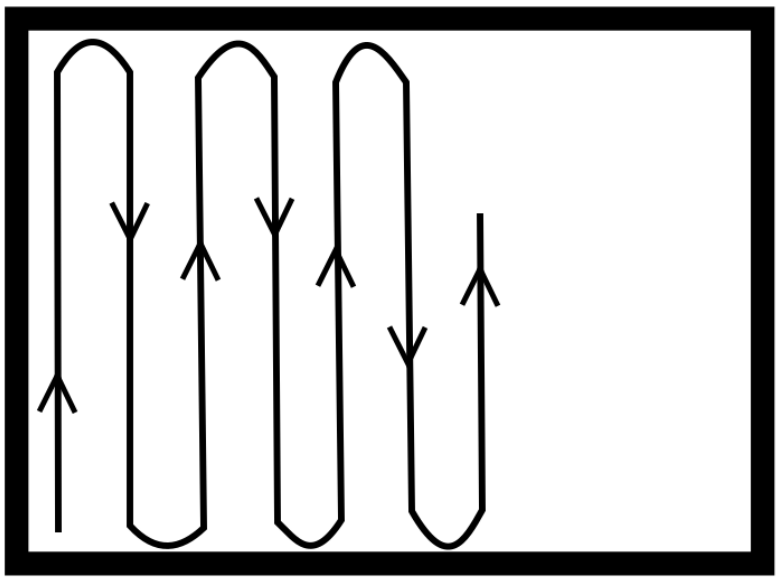
Second version

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	BI	Y	B	W	BI	W*	W*
Mbot Robot 2´4G																			
Motor 1	W*		1															W*	
Motor 2	W*		1																W*
Me Compass	W		1								W								
RJ25 cables																			
Structures and beams																			
Laptops			1																
Attrezzo (not essential)																			

ACTIVITY DESCRIPTION

First version

The robot should scan the area. The idea is that robot goes like that:

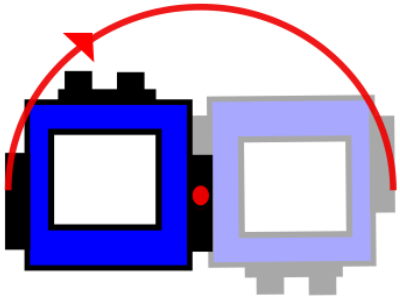


When the robot detects a black line it should make a 180° turn, but in such a way that it can avoid searching in the same area once again and not to miss too wide area. We also have to pay attention to the fact that a robot should be turning left and right alternately. That short description can be divided into following problems to be solved:

- Moving straight forward.
- 180° turn.
- Alternate turns.
- Turn enabling scanning the whole area, but avoiding repeating the route .

Problem: Scan the hole area

The last problem seems to be the easiest to solve. To turn right - the left wheel should go, and right wheel should stop.



Problem: Alternate turns

The next problem is that each time robot reaches the black line the turns change – robot turns: left, right, left, right,….

The list of steps:

STEP 1: K=0

STEP 2: Repeat until ultrasonic sensor >5

STEP 3: Move forward

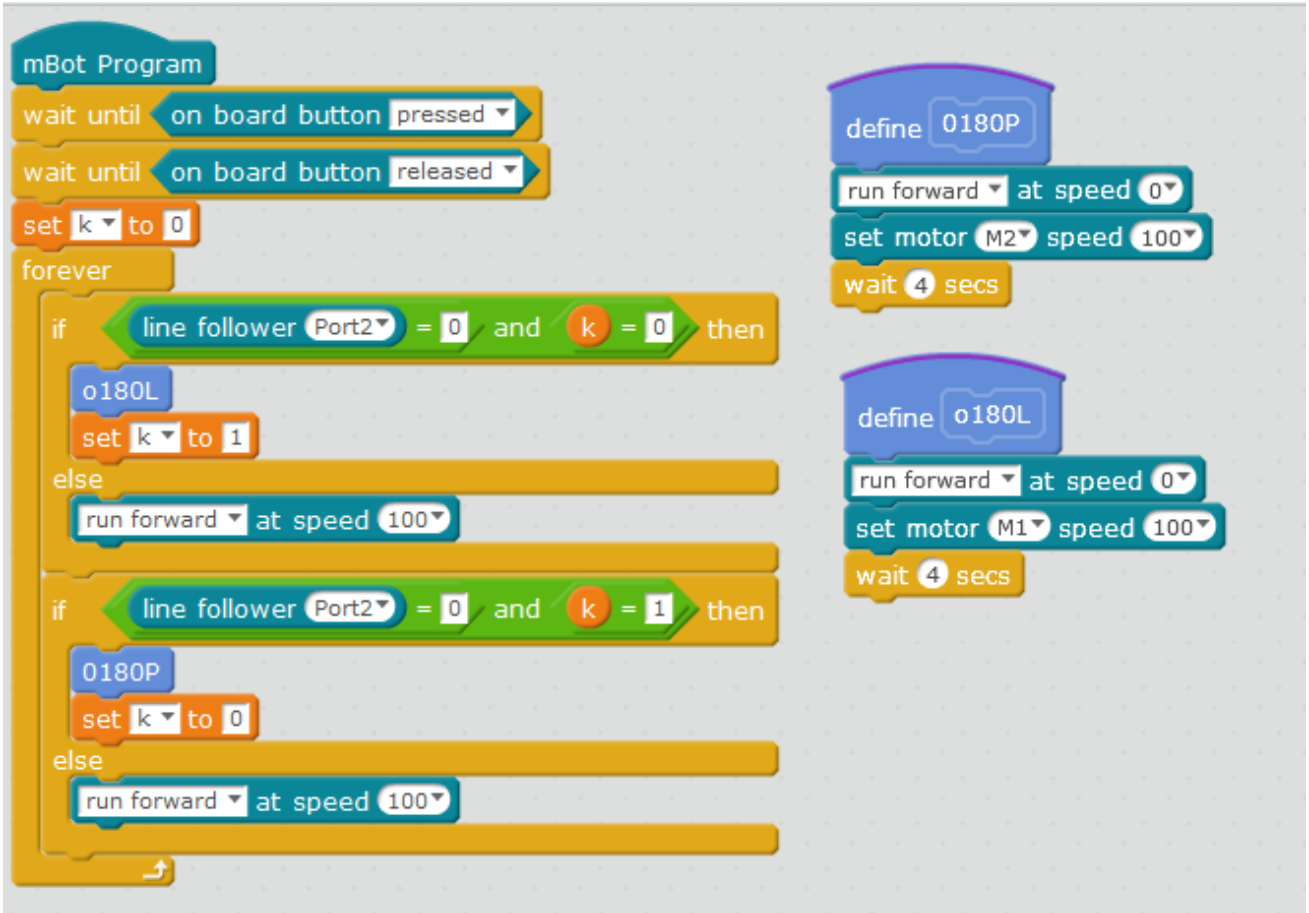
STEP 4: If line sensor = black and k=0

turn 180° left and k=1

or if line sensor = black and k=1

turn 180° right and k=0

or come back to Step 3



Problem: How to make a 180° turn precisely?

To define time of running It doesn't work well, because the speed depends on power of battery. In this project robot should turn with big precision.

We can use compass - we can tell the angle to change.

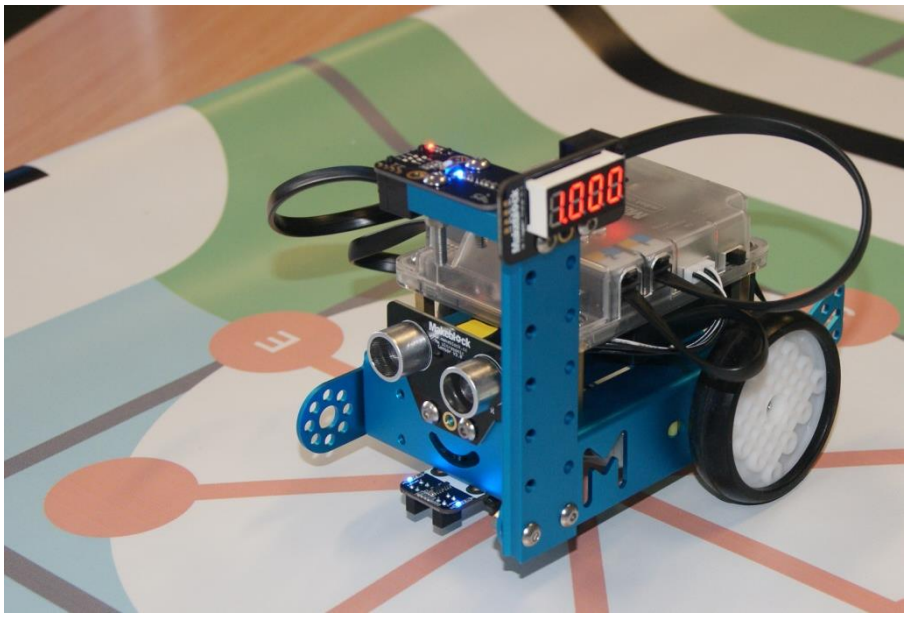
The compass

Compass is the sensor, which gives the number between 0 and 360.

0 or 360 means North. The number 90 means East, 180 – South, 270 – West.

Each program that uses compass should be used in Arduino mode. Before each use compass must be calibrated. During calibration sensor can sense some local interferences and adjust to the spot.

Mounting the compass:



Compass should be mounted in horizontal way. The X axis can be parallel to robot's axis. Compass is sensitive to ferromagnetic materials (iron or nickel), that is why you should try to mount it as high as possible. The robot's cover is made of aluminium that has an infinitesimal influence on the compass read-out, but the engines that produce an electromagnetic field are not neutral to its readings. Compass is also sensitive to posture – the more horizontal the mounting is, the more precise the readings are going to be. After careful mounting and calibration, we shouldn't accept measurement error higher than 2 degrees.

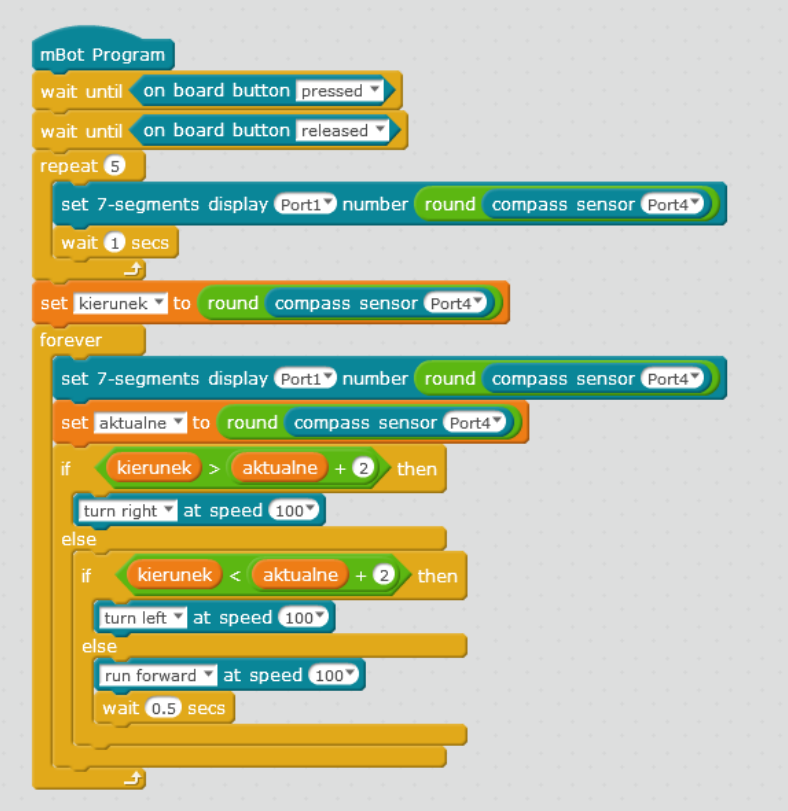
Calibration process:

Upload following program onto a robot.

Run the robot. Press the white button on the compass module – blue led should flash. You are in a calibration mode. Turn the robot slowly and smoothly right around its vertical axis by more than 360 degrees. During the turn the robot should be in horizontal position. The best idea is to put the robot on a big sheet of paper and to turn it round holding one of the corners. After the turn press the white button on the module again to exit the calibration mode. LED starts shining continuously.

Problem: going straight

Whenever we want the robot to go straight forward instead the usual 'go forward' block we can dub the procedure that at the beginning of the route is going to remember the track, and next of only the robot changes the set direction, a correction of the route is going to occur.

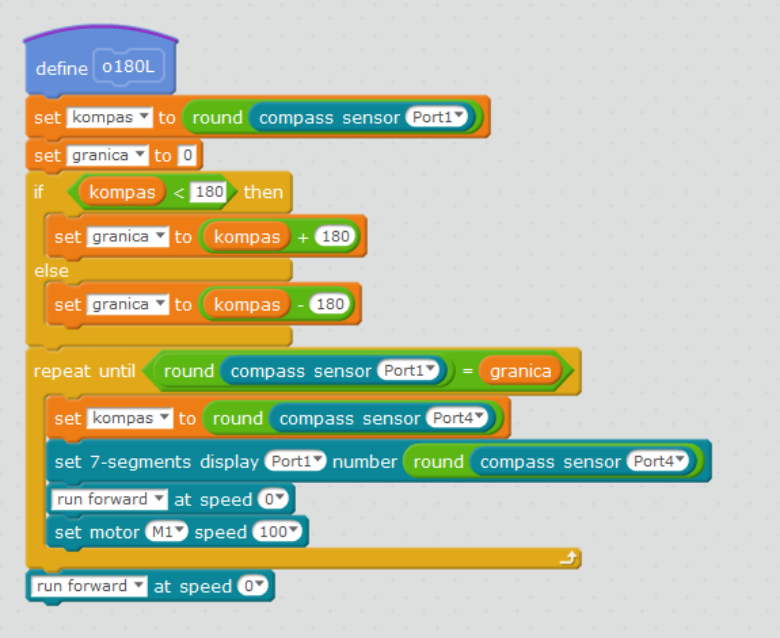


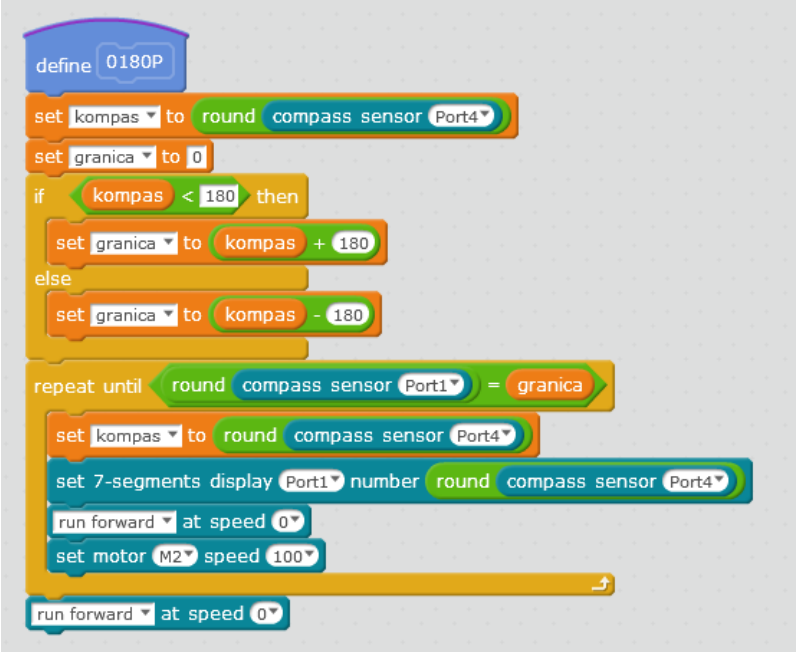
Problem: 180º turn

The next element of the program is 180 degrees turn. At the moment, when the robot overruns the black line, the current direction is saved. Nex,t the counter direction has to be found.

If the direction is less than 180, the counter direction is counted by adding 180. If the robot shows the number bigger than 180, the counter direction can be established by subtracting 180.

When we know that value, we tell the robot to turn until we get the proper compass reading.



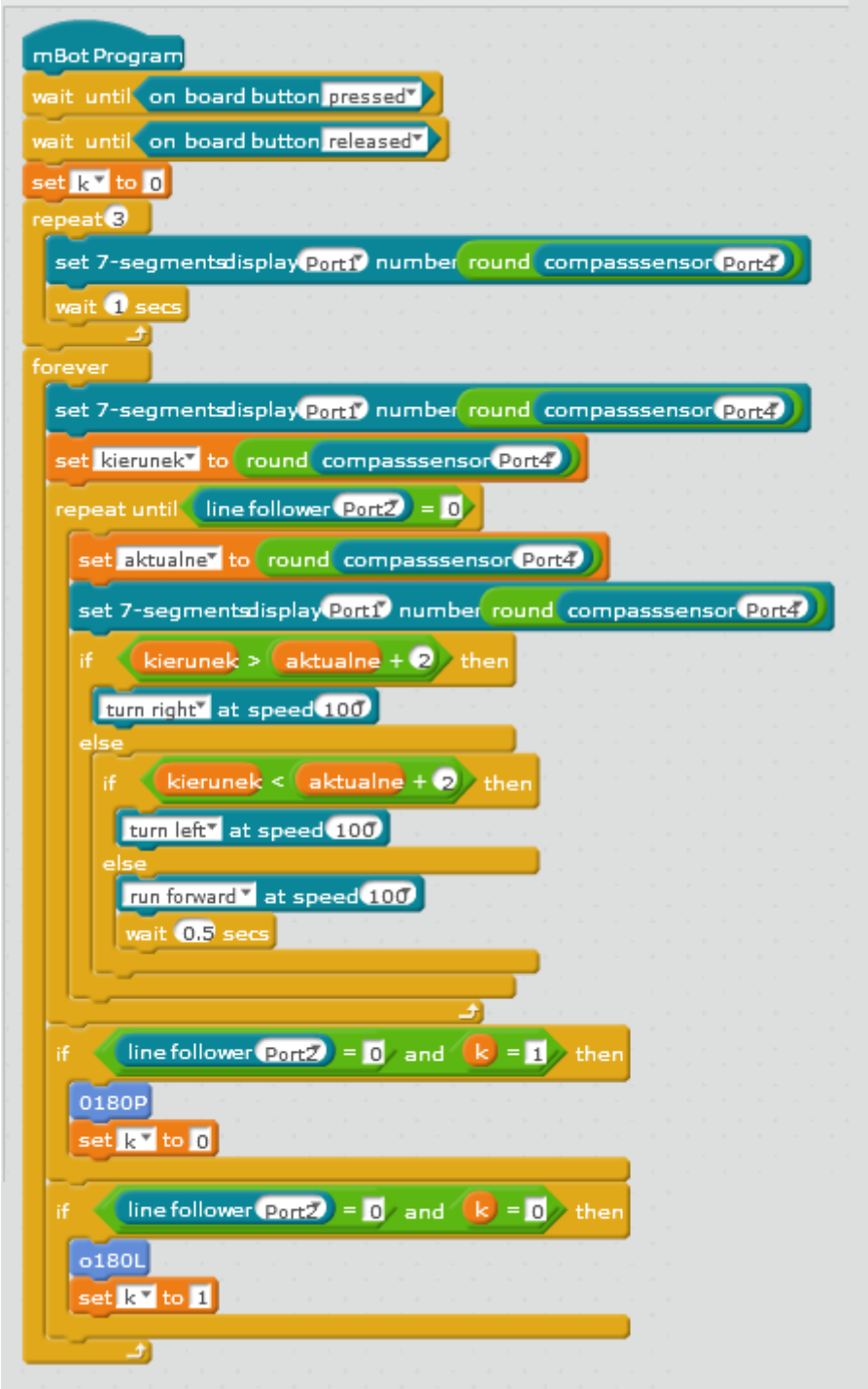


Main program

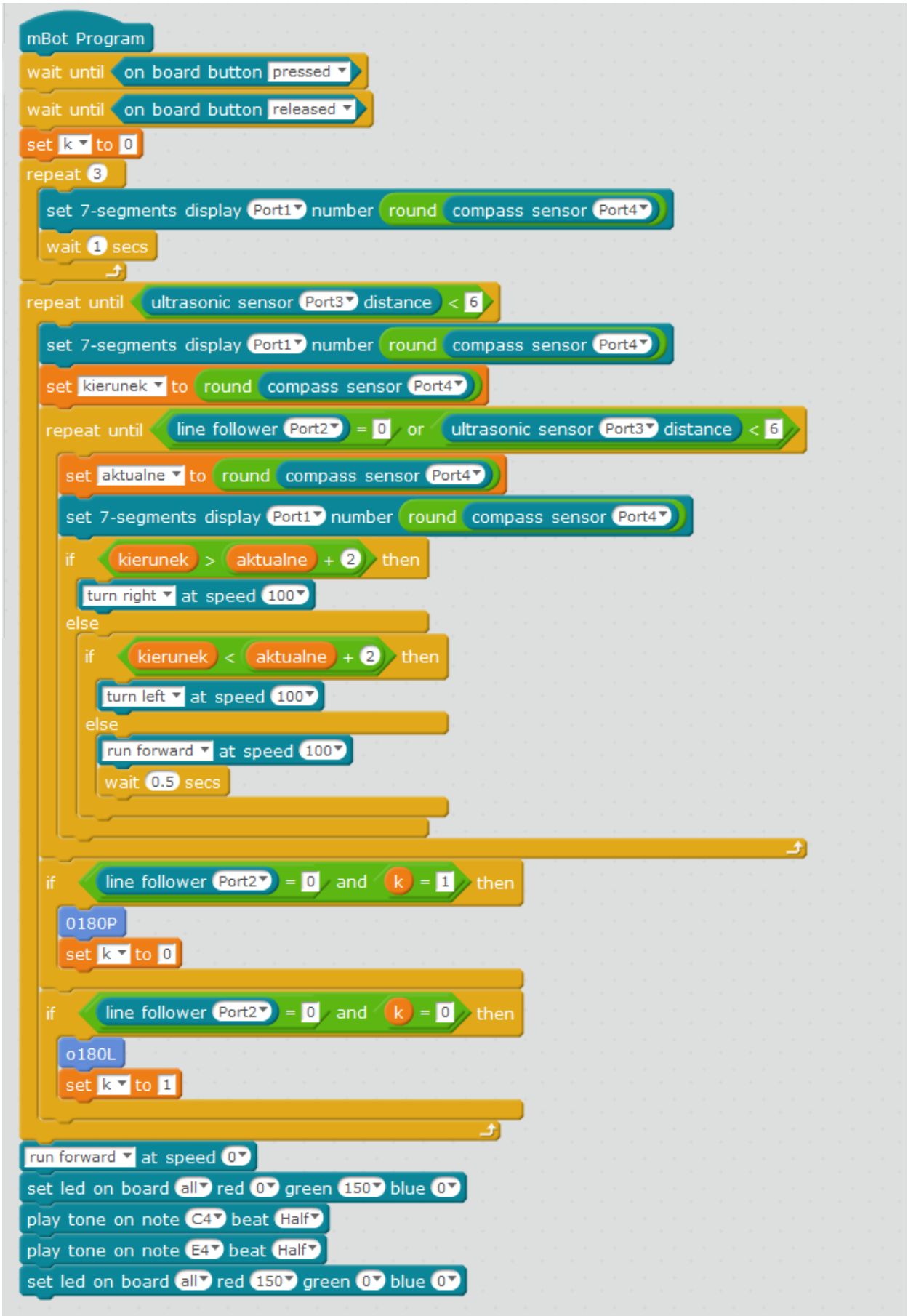
When we solved each of the problems separately, it's high time to put these elements together.

Program can be described as follows:

- Robot starts and goes straight according to the compass readings until it runs over the black line.
- When it runs over the black line, it turns right or left

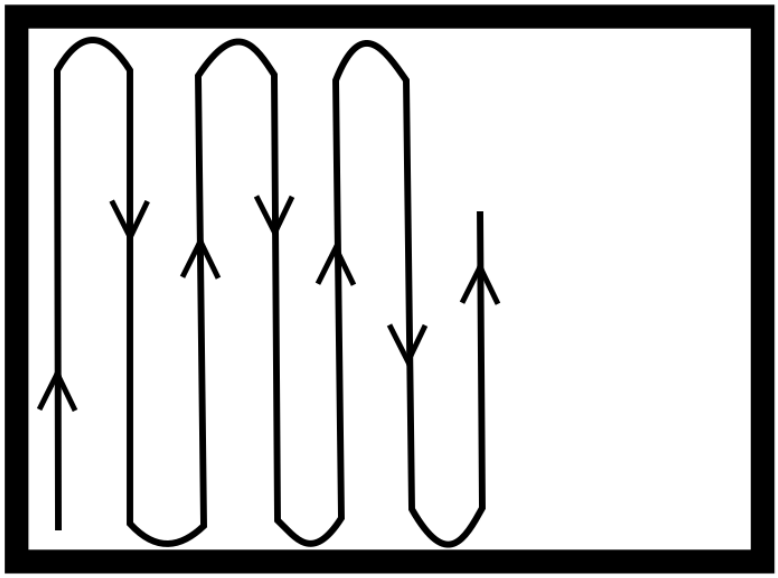


Now we have to add the condition, which stops the robot when it approaches the object being searched. Forever loop has to be replaced with Repeat Until. Additionally the like condition has to be made in the internal Repeat Loop.


mBot Program
wait until on board button pressed
wait until on board button released
set k to 0
repeat 3
set 7-segments display Port1 number round compass sensor Port4
wait 1 secs
repeat until ultrasonic sensor Port3 distance < 6
set 7-segments display Port1 number round compass sensor Port4
set kierunek to round compass sensor Port4
repeat until line follower Port2 = 0 or ultrasonic sensor Port3 distance < 6
set aktualne to round compass sensor Port4
set 7-segments display Port1 number round compass sensor Port4
if kierunek > aktualne + 2 then
turn right at speed 100
else
if kierunek < aktualne + 2 then
turn left at speed 100
else
run forward at speed 100
wait 0.5 secs
if line follower Port2 = 0 and k = 1 then
0180P
set k to 0
if line follower Port2 = 0 and k = 0 then
o180L
set k to 1
run forward at speed 0
set led on board all red 0 green 150 blue 0
play tone on note C4 beat Half
play tone on note E4 beat Half
set led on board all red 150 green 0 blue 0

Second version

The robot should scan the area. The idea is that robot goes like that:



The robot uses the compass to keep the right direction and waits for a fixed amount of time.

After that time it makes a 180° turn using the angle given by the compass.

By using angles is quite simple to obtain alternate turns.

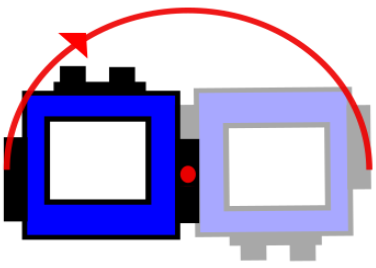
In order to cover a wide area all you need is more time to search so the user has to setup it in the mblock software.

The algorithm is composed by four logical steps:

- Moving straight forward.
- go straightforward for “X” seconds.
- 180º turn .
- Alternate turns.
- Turn enabling scanning the whole area, but avoiding repeating the route .

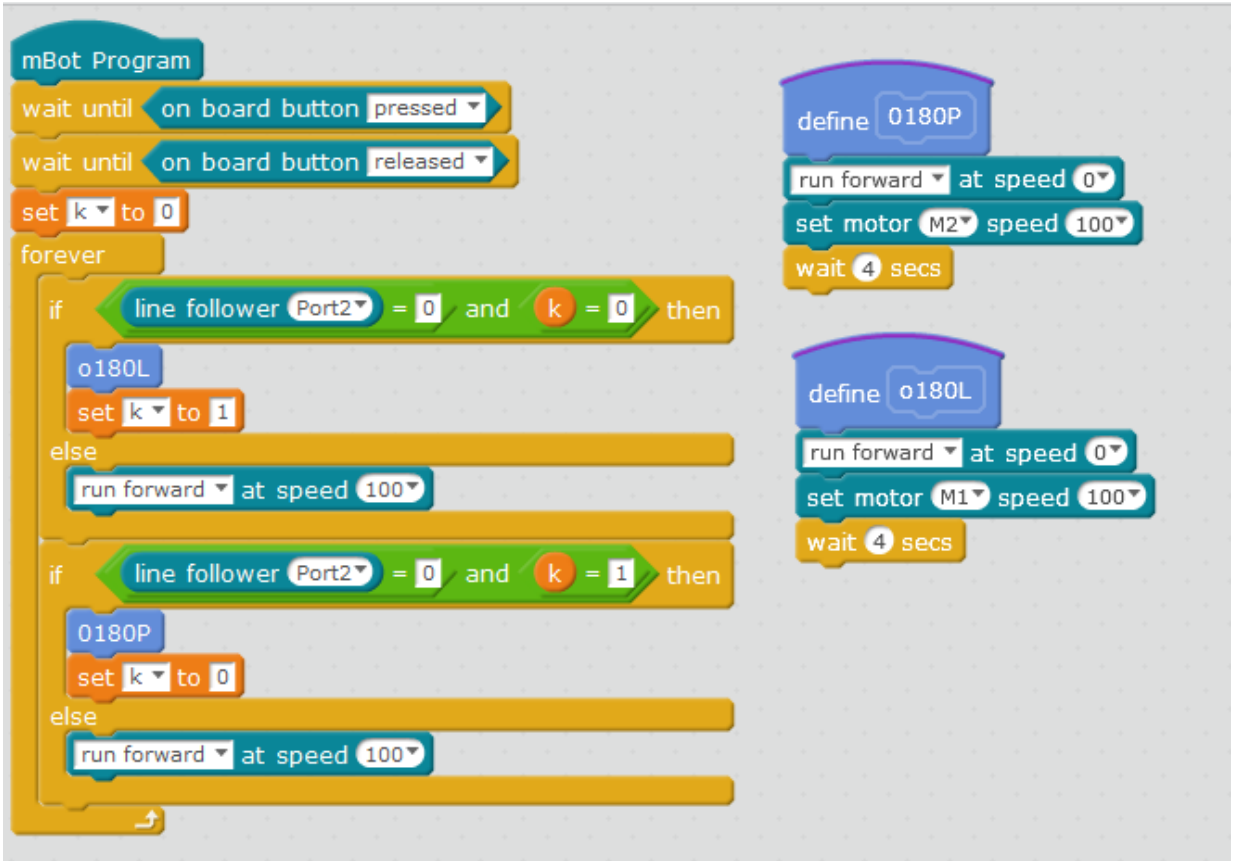
Problem: Scan the hole area

The last problem seems to be the easiest to solve. To turn right - the left wheel should go, and right wheel should stop.



Problem: Alternate turns

We solved this problem using a simple counter in the software



Problem: How to make a 180° turn precisely?

We can use the compass and measure the angle.

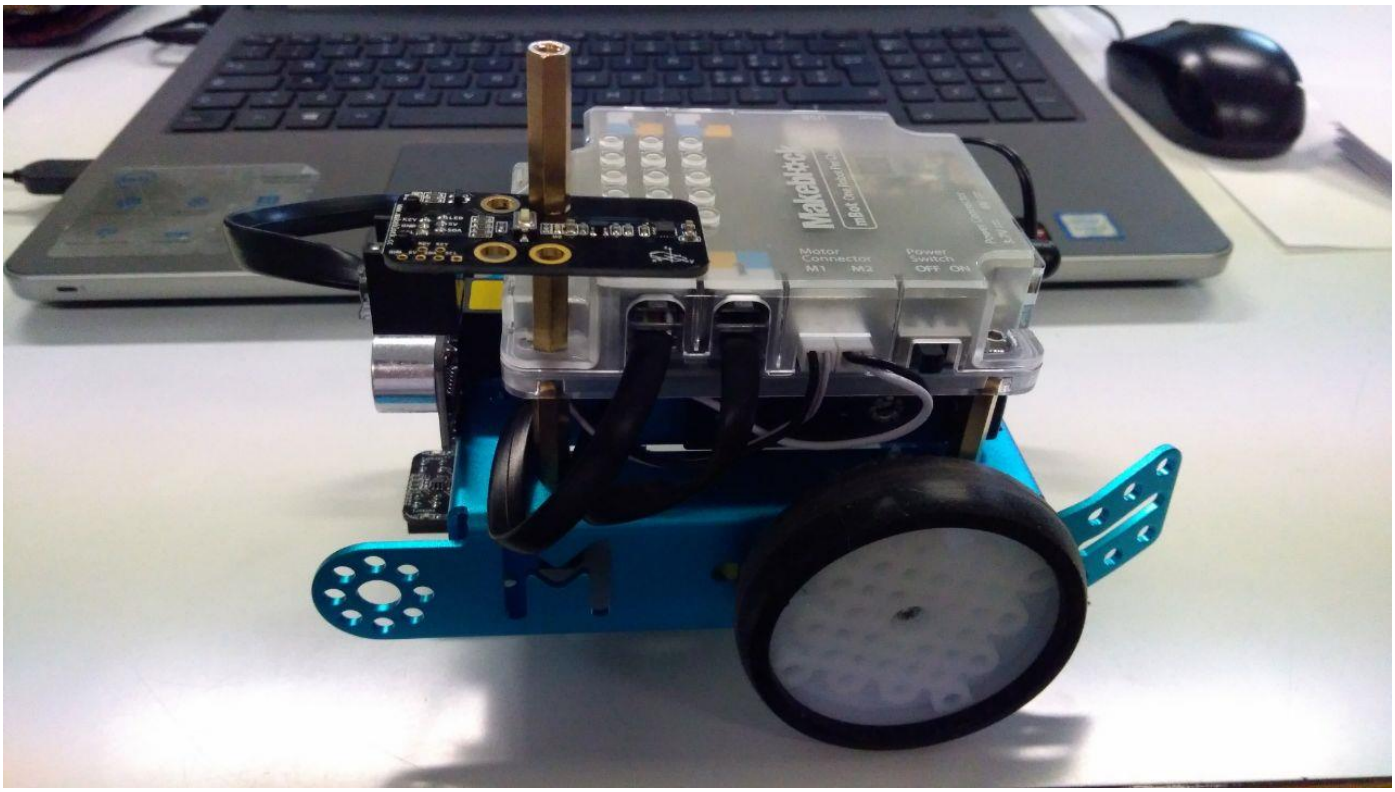
The compass

Compass is the sensor which gives the number between 0 and 360.

0 or 360 means North. The number 90 means East, 180 – South, 270 – West.

Each program which uses compass should be used in Arduino mode. Before each use compass must be calibrated. During calibration sensor can sense some local interferences and adjust to the spot.

Mounting the compass:

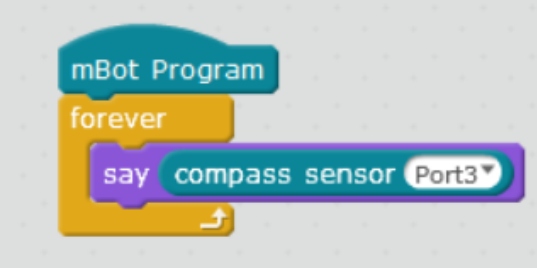


Compass should be mounted in horizontal way. The x axis can be parallel to robot's axis. Compass is sensitive to ferromagnetic materials (iron or nickel), that is why you should try to mount it as high as possible. The robot's cover is made of aluminium that has an infinitesimal influence on the compass read-out, but the engines that produce an electromagnetic field are not neutral to its readings. Compass is also sensitive to posture – the more horizontal the mounting is, the more precise the readings are going to be. After careful mounting and calibration we shouldn't accept measurement error higher than 2 degrees.

Calibration process:

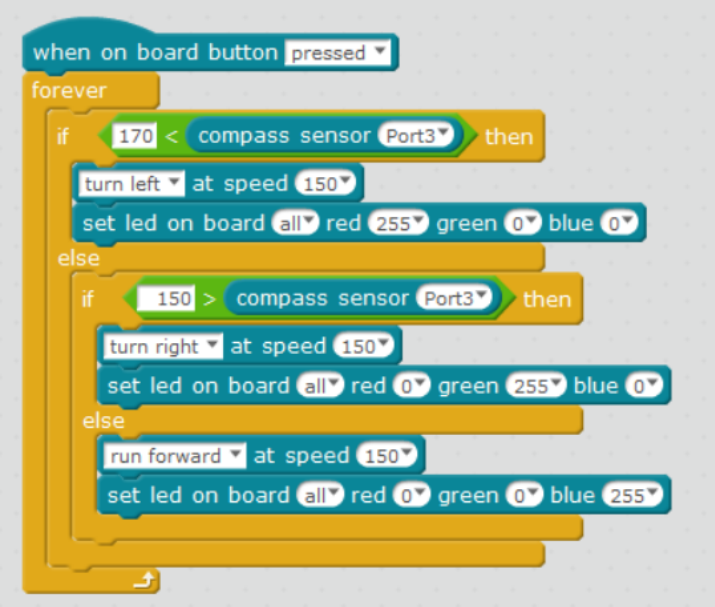
The best way to calibrate the compass is to position the mbot on a sheet of paper and, as shown at the beginning of the video, turn the mbot on itself.

In order to see the angle measured by the compass, connect the mbot to the computer and use the first software attached "0_compassOnly.sb2".



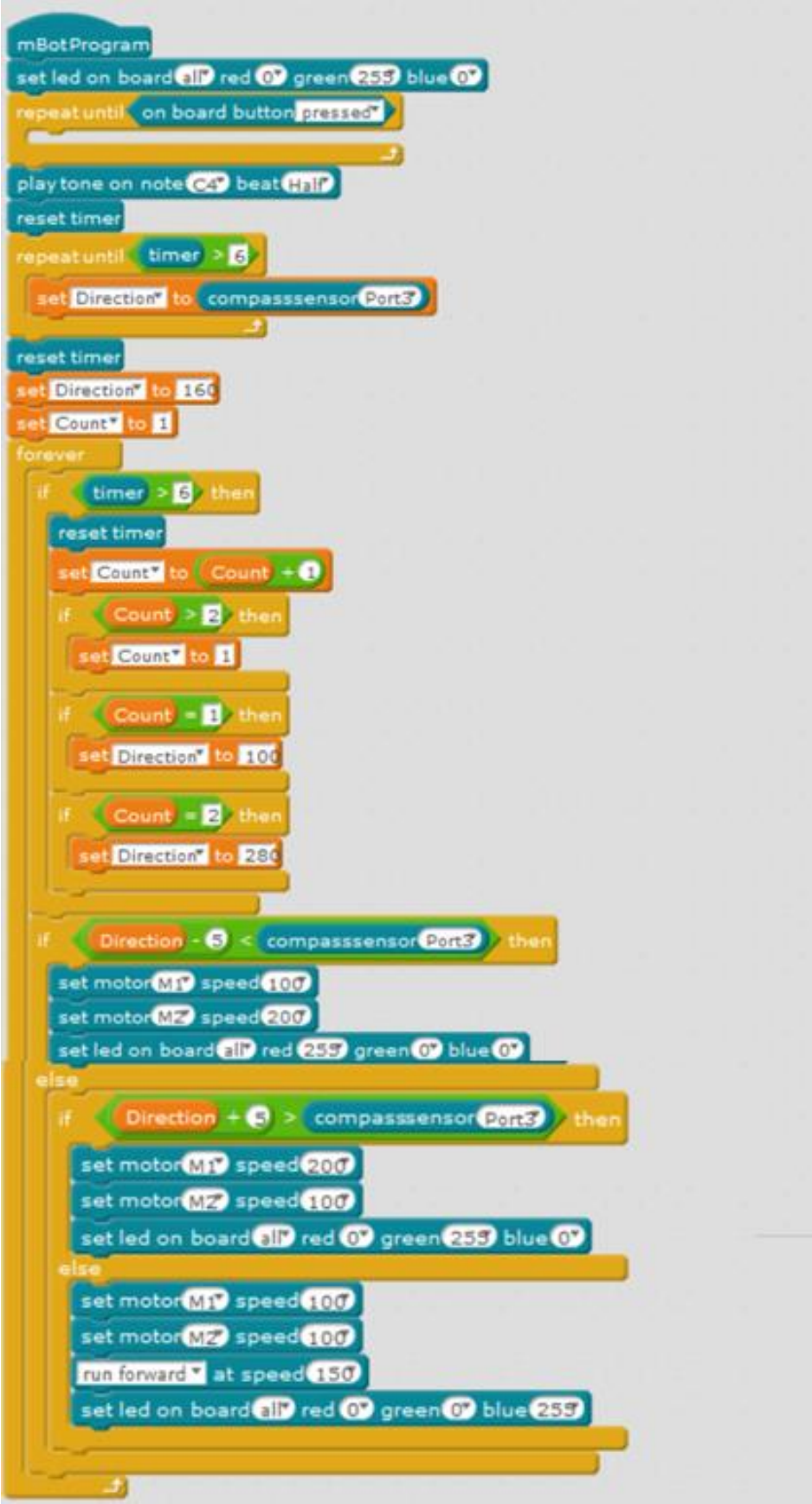
Test the compass and the Mbot movements

Once you have tested the compass alone and found that it works fine you can cable it on the mbot and try the following software. In this way you know that the mbot catch the signal from the compass and it can change direction as expected. Look at the video to see how the mbot works with this software.



Main program

Keeping all together we obtain the complete software in order to cover the complete area.



STUDENT'S EVALUATION

After this activity student can:

- Explain how to use the compass.
- Interpret the readings from compass.
- Use compass to keep direction of robot.

BIBLIOGRAPHY

https://trobot.pl/sklep/czujniki-moduly/makeblock-kompas/

http://learn.makeblock.com/en/me-compass/

SCALABILITY

Compass can be used to teach elements of geography – students use analog compass to compare the readings.