# MRI Fusion Robots For Beginners

## Programming with Blockly

(Teacher/Mentor Web/USB Edition 1.48 - November 2018)

Ying Chen, Yaya Lu and DrGraeme

*https://www.DrGrae.me*

# Introduction and Overview FB1

This is a "first look" at Modern Robotics Inc.'s Fusion Robot, using pre-release software. The current version (v1.0.5) of Fusion's operating system is a little different, but we document the differences. While these tutorials have less detail than our usual "Absolute Beginners" courses, we aim to include sufficient detail to enable use by a home-study parent/child combination, or within a class by a class-teacher who has limited technical experience. For this audience, we emphasize detailed "how-to" videos, and minimize "talking-head" presentations. We introduce new concepts only when they are needed to continue using our Fusion Robot.

What does this course cover?

- This introductory course *does* cover the start of the use of Blockly with the Fusion Robot Base Kit. We also check out some sensors that are available for use with Fusion, but which are not included in the Basic Fusion Base Kit; these tutorials are labeled "Extras".
- This course *does not* cover either the complete use of Fusion Blockly, or the use of Python which is the alternative computer language that can be used with Fusion. We are currently preparing a Python course for Fusion, but it is not yet (November 2018) available in a completed form.

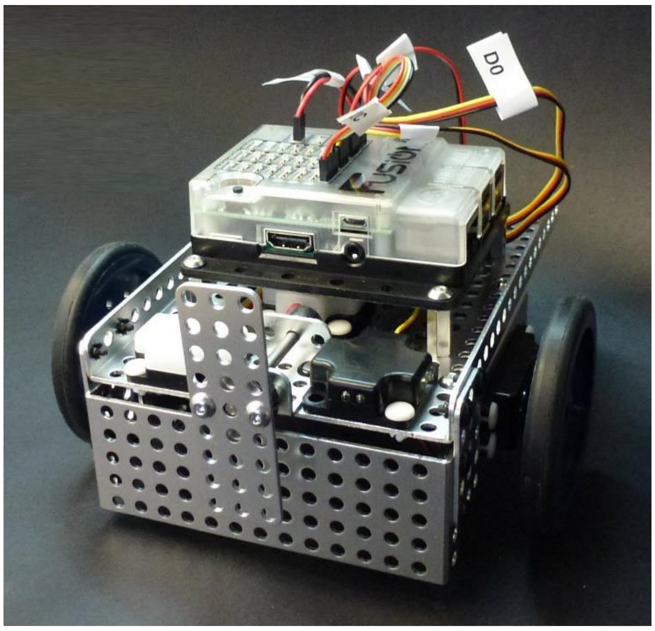
What should you do next?

- If you are a student interested in building your Fusion Robot, turn to the next page and start!
- If you are a teacher or mentor interested in a semi-technical summary of your Fusion Robot's features, turn to the "Technical Extras" on page 28.
- If you have a Spartan Robot and wish to change to a Fusion Controller, go to page 29.

**Note:** If you have no Internet access and are using our USB videos (available soon), the Web locations referenced in these notes *will not* be available, but our own Fusion videos *will* be available.

Enjoy! 😉

# Building Your Fusion Robot FB2



## ***Unpacking Fusion's Mailing Box*** FB2a

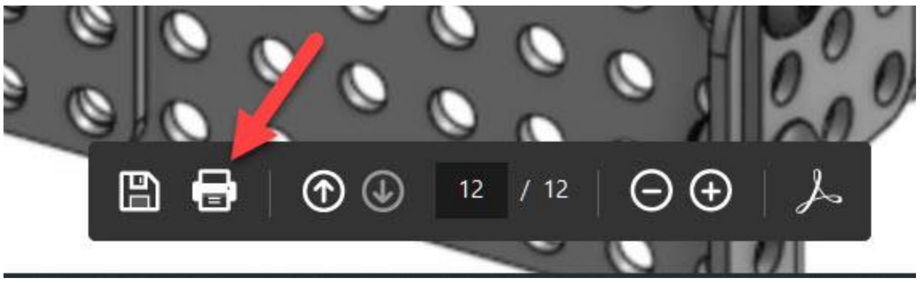
One of the first things to do is to check the contents of the mailing box you received. In the video below we go through the components of our mailing box, to see if they are the same as those listed on Modern Robotics' *Fusion Base Bot Building Instructions* pdf that you can see by clicking here.

Video - Web - USB

## ***Either: You Prefer Printed Build Instructions for Fusion?*** FB2b

Modern Robotics provides printable instructions for building your Fusion Robot, click here to see them.

If you prefer to build your Fusion Robot from a paper copy of these build instructions, for most browsers I have tried (Chrome, FireFox, Opera, Brave, Bing) there will be a little printer image on the "build instructions" screen on which you can left-mouse-click to print these instructions. If you are using the Internet Explorer browser, you may have to left-mouse-click towards the bottom of the "build instructions" screen, and the window below will appear:



Click where the arrow points to print out these build instructions.

***Errata?*** Note that on the second-last page of these build instructions, the two last steps should be "Step 13" and "Step 14" (November 2018).

## *Or: You Prefer Video Build Instructions for Fusion?*

Modern Robotics has not yet (November 2018) provided a YouTube video demonstrating a Fusion build. However, their Spartan Robot has a similar chassis to the Fusion Robot, and thus their YouTube video of a Spartan build can provide some assistance if you want more help than is provided by the printed Fusion build document.

Modern Robotics' Spartan complete build video can be found by clicking here.

Portions of this Spartan video that correspond to steps in the printed Fusion build document are listed below. We will re-use these steps to illustrate the Fusion build.

**Fusion Step 1:** Assemble base plate. Web

**Fusion Step 2:** Add battery mount and back "wheel". Web

**Fusion Step 3:** Place the standoffs in the hole locations shown in the Fusion build document step 3, NOT the hole locations visible in the Spartan video. Later, in Step 14 of this Fusion build, your Fusion controller will be mounted on the top of these standoffs. The Fusion controller is physically bigger than the controller used in the Spartan, this is why different positions for the standoffs are needed. Web

**Fusion Step 4:** Add motors. Don't worry about the Spartan controller shown in the video – we will add Fusion's controller later in Step 14. Web

**Fusion Step 5:** Assemble the front plate. Web

**Fusion Step 6:** Attach the front plate to Fusion. Note that, compared with the Spartan build, the Fusion Build document reverses the 5x13-hole flanged plate so that it is "concave" when viewed from the front of the Robot. We don't know why Modern Robotics have made this change. We don't like it. We would prefer that this plate to be mounted in a "convex" position when viewed from the front, just as it is in the Spartan build. This will allow us in the "Wall Follow" tutorial (page 21) to mount our Optical Distance Sensor sideways on the front plate, to demonstrate how to teach our Fusion Robot to follow a wall. Web

**Fusion Step 7:** Add the wheels. Web

**Fusion Step 8:** Add the battery. Web

**Fusion Step 9:** Add the touch sensor. Web

**Fusion Step 10:** Build the touch sensor sub-assembly. Web

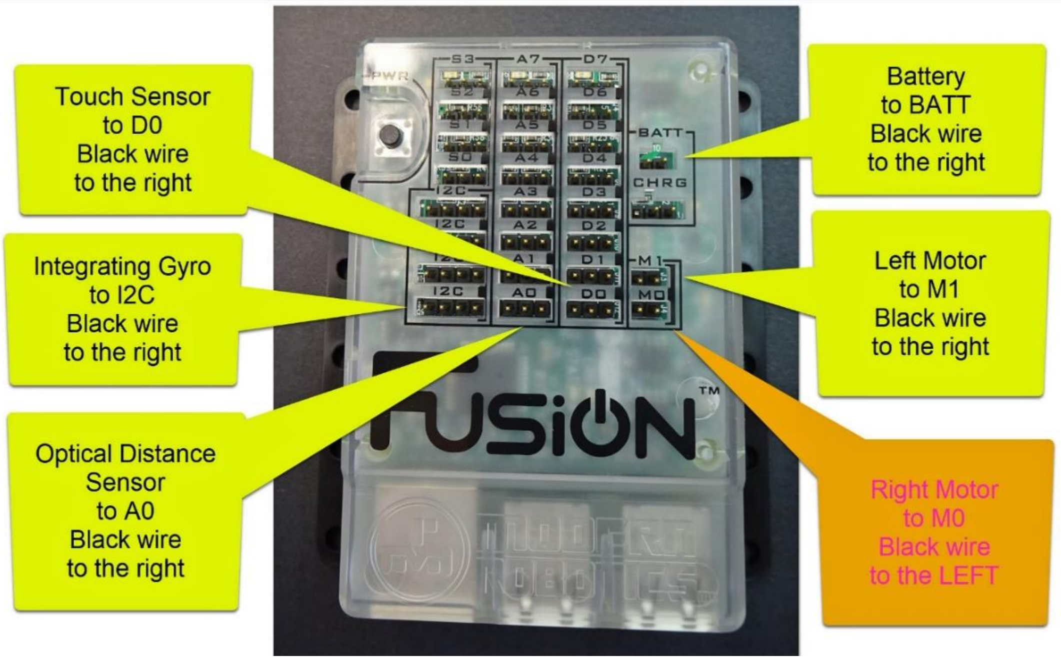
**Fusion Step 11:** Attach the touch sub-assembly to Fusion. Web

**Fusion Step 12:** Viewing from the front of the Robot, Fusion's single Optical Distance Sensor is mounted on the right side of the Fusion robot. Web

**Fusion Step "13":** Viewing from the front of the Robot, Fusion's single Gyro Sensor is mounted on the left side of the Fusion robot, in the same place as the left sensor in this video. Web

**Fusion Step "14":** Attach the Fusion controller to the standoffs we added in Step 3. Web

**Fusion last page:** Next carefully connect the cables in to the Fusion controller, as shown below.


Touch Sensor to D0 Black wire to the right
Integrating Gyro to I2C Black wire to the right
Optical Distance Sensor to A0 Black wire to the right
Battery to BATT Black wire to the right
Left Motor to M1 Black wire to the right
Right Motor to M0 Black wire to the LEFT
FUSION

## *Now Charge Fusion's Battery Pack.*

It is unlikely that there will be sufficient power in Fusion's battery pack to allow you to experiment with your Robot. Turn Fusion off, because the battery will not charge when Fusion is turned on. Connect the charger supplied as part of your Fusion kit in to your building's power supply, and plug the other end in to Fusion's slot CHRG. The black wire can be either to the right or to the left, both positions will charge Fusion's battery. If the battery is flat, the light on the charger will glow red. It will take an hour or two to charge Fusion's battery pack. When the battery is charged, the light on the charger will change to a green glow.

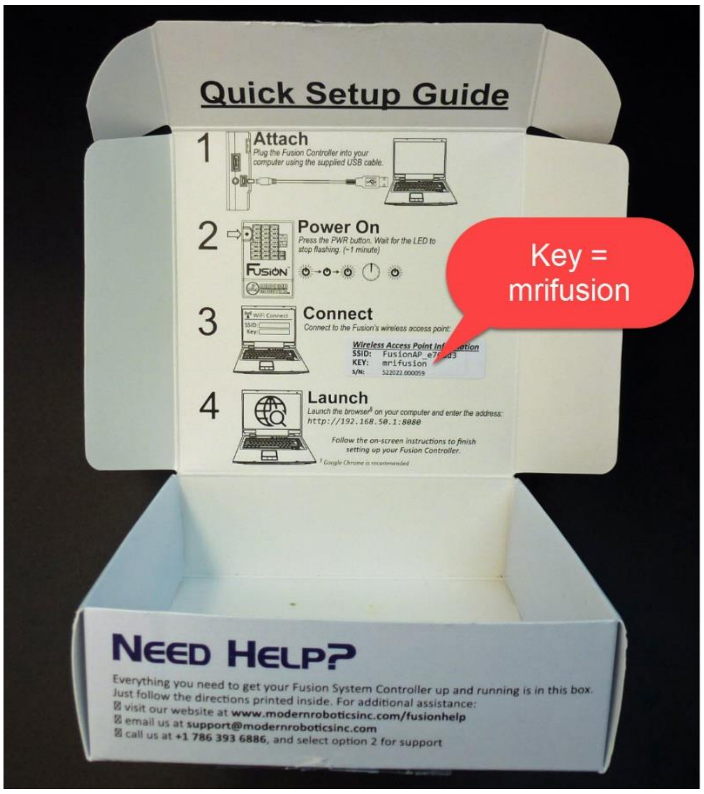
***NOTE:*** *Make sure Fusion is turned off.* If Fusion is turned on, and you plug in the charger cable, the light built in to the charger will glow green (erroneously indicating a full battery) when in fact the battery could be almost flat. *Your Fusion battery will only be charged when Fusion is turned off.*

### *Congratulations!*

You have now assembled your Fusion robot. In the next lesson you will find out how to connect your laptop to your Fusion robot using Wi-Fi. Let's go…

## Connect to a PC using Fusion's Wi-Fi FB3

The first time you connect your computer to your Fusion Robot, you may be required to provide a security key. This security key will be supplied by Modern Robotics Inc. as part of your Fusion kit. At the time of writing (October 2018), this key is mrifusion. It can be seen inside the lid of the Fusion System Controller Box that comes with your Fusion kit.



Quick Setup Guide

1 Attach
Plug the Fusion Controller into your computer using the supplied USB cable.

2 Power On
Press the PWR button. Wait for the LED to stop flashing. (~1 minute)

3 Connect
Connect to the Fusion's wireless access point:

Wireless Access Point Information
SSID: FusionAP_e70dd3
KEY: mrifusion
S/N: 522022.000059

4 Launch
Launch the browser§ on your computer and enter the address:
http://192.168.50.1:8080

Follow the on-screen instructions to finish setting up your Fusion Controller.

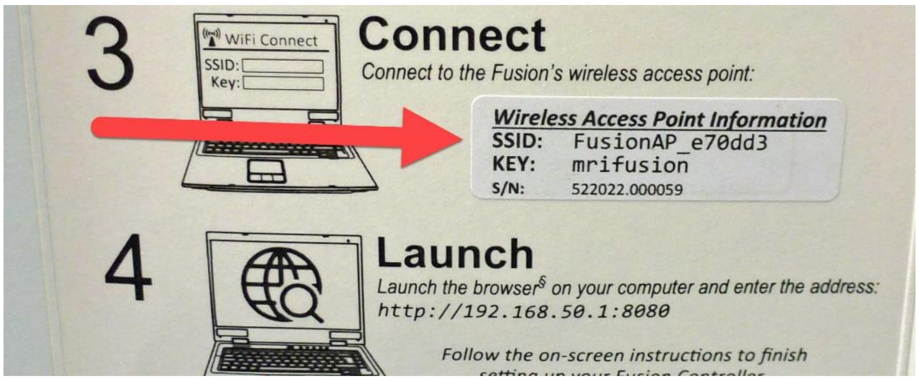
§ Google Chrome is recommended

NEED HELP?
Everything you need to get your Fusion System Controller up and running is in this box. Just follow the directions printed inside. For additional assistance:
- visit our website at www.modernroboticsinc.com/fusionhelp
- email us at support@modernroboticsinc.com
- call us at +1 786 393 6886, and select option 2 for support

This video below shows how to use Fusion's inbuilt Wi-Fi to link a Fusion Robot to a computer. In this example, the computer is a Windows 10 PC.

Video - Web - USB

Modern Robotics also have information about connecting your computer to your Fusion Robot via Macintosh, via Linux, via iOS and via Android. You may need the SSID of your Fusion Robot, which can be found here.



3 Connect
Connect to the Fusion's wireless access point:

Wireless Access Point Information
SSID: FusionAP_e70dd3
KEY: mrifusion
S/N: 522022.000059

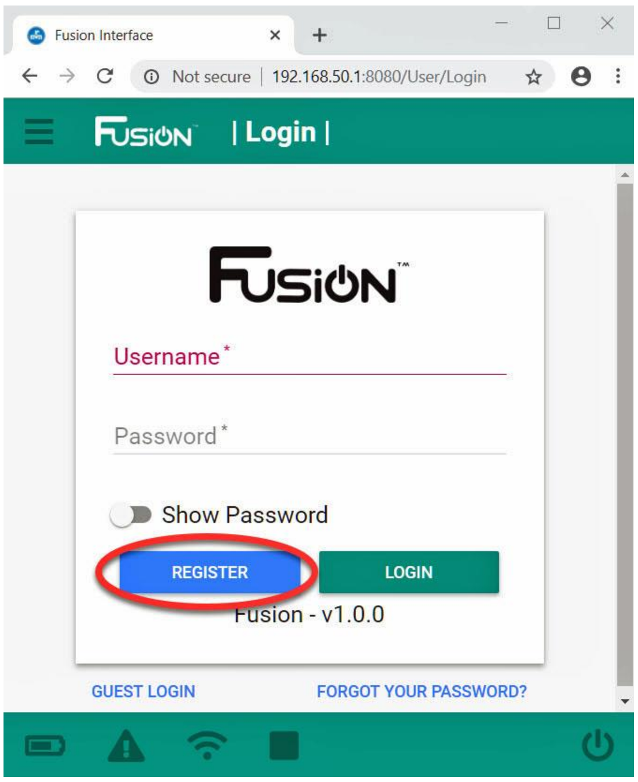
4 Launch
Launch the browser§ on your computer and enter the address:
http://192.168.50.1:8080

Follow the on-screen instructions to finish

One of the unique things about a Fusion robot is that, as well as acting as a robot, it also acts as a server with an inbuilt Wi-Fi link. Laptops, desktop PCs, tablets or smart phones can connect to the Fusion robot using this Wi-Fi link. It is almost as if the Fusion robot is its own Wi-Fi hotspot, except that the Wi-Fi connection is not to the Internet, it is just to the Fusion robot itself. In normal operation, neither the Fusion robot nor the computer is connected to the Internet, (but either can be if the student *really* desires this to be the case).

This ability to be able to operate without a connection to the Internet can be a major advantage when the Internet connection is a problem, either because of slow Internet speed or because the Internet is simply not available in the student workplace. It can also be an advantage when a school is concerned that the student may have contact with some of the less desirable content on the Internet, and therefore has banned connections between student computers in the school, and the Internet. In both these circumstances, the Fusion robot can be used and programmed independently of any Internet access.

# Setting up Admin and User Accounts in your Fusion Robot FB3b



One of the advantages of the Fusion Robot is that multiple accounts can be set up inside the one Fusion Robot. Thus, students in many classes can all (one at a time) use the same Robot, each student saving their programs inside the Fusion Robot. Students are subsequently able to login in to Fusion again later, and continue their work, starting from where they left off. This makes it very easy to use a Fusion Robot across multiple classes.

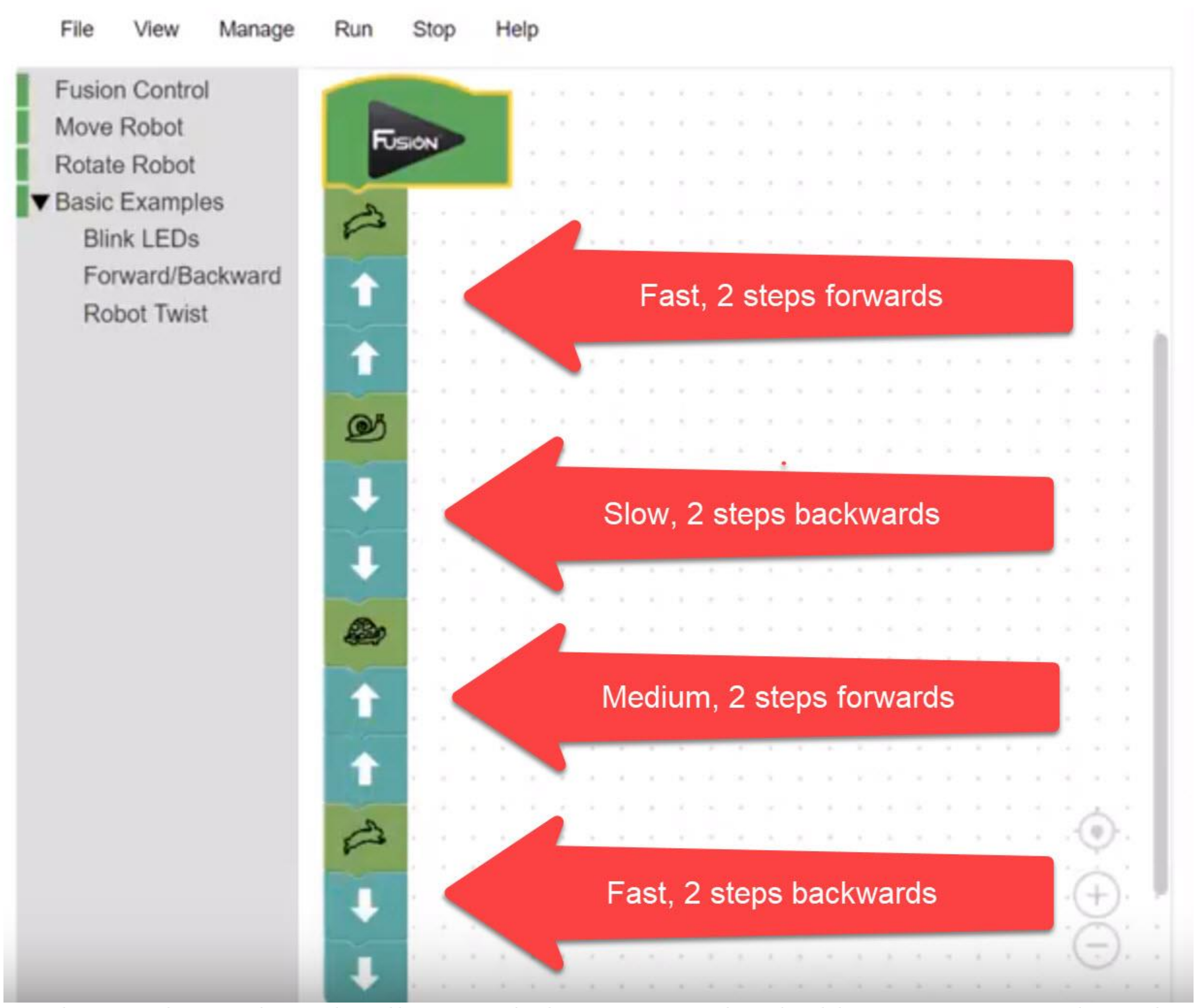
However, before Fusion can be used by multiple users, accounts within the Fusion Robot must first be set up. There are two types of Fusion accounts, "Admin" and "User". When used in a classroom, usually only the teacher or mentor will have access to the more powerful "Admin" account. Students would usually use a "User" account. The video below demonstrates how to set up both types of accounts within a Fusion Robot.

Video - Web - USB

***Note 1:*** *Some of the button colors in this tutorial may not match the button colors you see when you set up your accounts in your Fusion robot. This difference has occurred because Modern Robotics has updated their Fusion operating system since this video was produced. The method demonstrated in this video will still work well with the new version of Fusion's operating system.*

***Note 2:*** *Modern Robotics have recently made available web documentation of this process, which can be accessed by clicking here.*

# Getting Fusion to Move FB4



Modern Robotics have kindly provided some sample Blockly programs. In the video below we show how to run one of the example programs that are contained within our Fusion Robot.

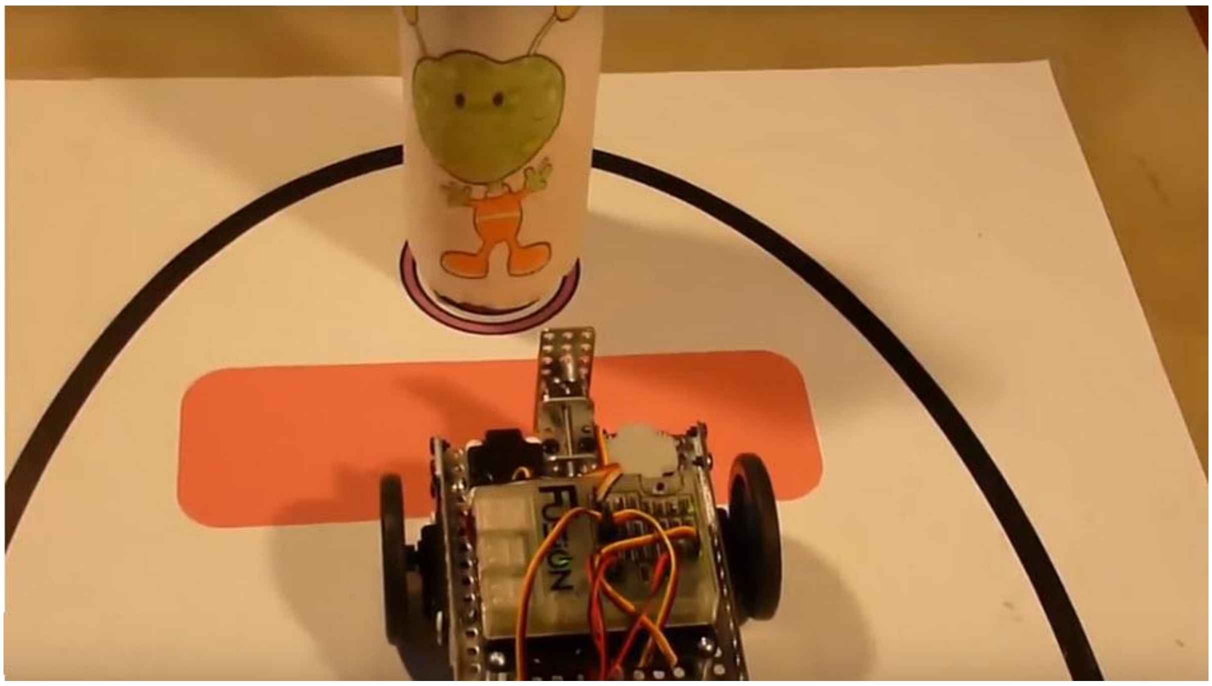
It is assumed that a Wi-Fi link between your computer and your Fusion Robot has already been set up (if you have not done this, look at our previous tutorial). In this video we start up a browser (theoretically any browser, but Modern Robotics recommend Google Chrome, so that is what we will use). We link into Fusion, look at one of Modern Robotics' example Blockly programs, and run it. This video shows the effect of our first run using this sample Blockly program to control our Fusion Robot.

***Note:*** In this video I use an "Admin" account. This tutorial will work just as well if a "User" account is used.

***Update:*** In the latest version of Fusion software, the example program has been moved to menu item: Help/Examples/Basic/driveForwardBackward.blk

Video - Web - USB

# Fusion Approaches an Alien! FB5



Let us pretend that an Alien Ambassador has come to Earth. We don't know whether the Alien is dangerous or not. We will send our Fusion Robot to approach the Alien instead of us, because if anything goes wrong it will be a Robot and not us that will be zapped. We need to teach Fusion to move to the "red carpet", pause, and retreat (backwards -it is impolite to turn one's back on an Alien Ambassador!) Do not let Fusion collide with the Alien Ambassador – that might start an intergalactic war!

***Note:*** In this video I use an "Admin" account – this tutorial will work just as well from a "User" account.

Video - Web - USB

If you have access to an A1 printer, you could download and print out this arena (.pdf format) that we use.

Download – Web - USB

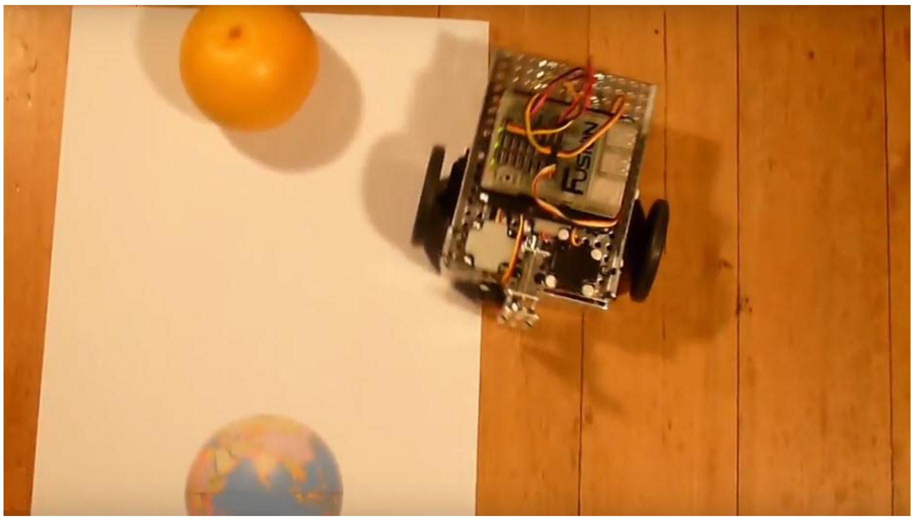
A second similar arena, also in pdf format but using slightly different colors, is also available for download:

Download – Web - USB

Making an "Alien" has been a delightful project in some classes in which we have assisted. If time is short, an image that we have used to wrap around a standard PET drink bottle is:

Download – Web - USB

# Fusion Goes Around the Moon FB6



"Around the Moon" is a Challenge we have used with Middle School students. The Robot has to leave a pretend "Earth", go into "space" around the (orange in this case!) moon, and land back right on top of the "Earth".

If you have access to an A3 printer, an A3 sized "Moon/Earth" image suitable for use in this lesson is available for free download – Web – USB; otherwise click here - Web - USB.

## *Fusion goes around the Moon using Basic Blockly Code*. FB6a

The video below shows that a solution to this Challenge is possible using only the *Basic Blockly* commands. It demonstrates that a tutorial could be written for this Challenge using only the simplest level of commands available in Fusion's Blockly.

Video - Web - USB

## *Fusion goes around the Moon using Intermediate Blockly Code*. FB6b

Again, we will send our Fusion Robot around the (orange) moon. Up until now, we have been using "Beginner Blockly" commands. In Modern Robotics' Blockly system, we also have access to more detailed "Intermediate Blockly" commands. These will allow us to have more detailed and accurate control of our Fusion robot. The video below demonstrates how to use some of these commands to obtain a solution to our *Moon Challenge*.

Video - Web - USB

## *Extensions to the "Around the Moon" Challenge.*

Why do we consider extensions to this Challenge?

When teaching using robotics, right from the start with the Tasman Turtle in the early 1980s, through other robots including LEGO NXTs & EV3s over the last decade, we observed our students exhibiting a tremendous range of abilities. To help keep our students stimulated, we use "extensions". We could do this because our tutorials are not of the "lock step" type that occur so frequently, particularly in Secondary School classes.

Our Challenges are carefully designed so that the slowest student in the class can achieve a successful result, and not be left behind. This allows us to come very close to not having any students left feeling a "failure" in our classes – every student succeeds at a level where they may be encouraged.

In this "Moon Challenge", even the slowest student has (so far) ended up with a robot ending up quite close, or perhaps with a tiny bit of the robot touching some part of the "Earth", which would be counted as a successful completion, with the student being legitimately congratulated on their good result.

With average students, we would try and encourage them to get their robot completely over the "Earth", so that the Earth can not be seen when looking at the robot from above. This is quite a bit more difficult and requires more time, which is a suitable target for average students, while the slower students complete their less difficult version of this Challenge.

Gifted or talented students will have generally completed this Challenge faster than the average students in the class. To save them from getting bored and frustrated while the rest of the class catches up, we have extensions of these Challenges that will give them more difficult practice in what we are trying to teach them. Our experience over the roughly the last decade with LEGO NXT and EV3 robots, is that this approach will keep gifted students happily continuing to learn, while the slower students succeed in their less elaborate versions of the Challenge.

In the video below, we use what one Grade 5 student called his "Sufi" approach to "Going Around the Moon". He had read about the Sufi Whirling Dervish dancers who spin around during their dances (see here). He wanted his robot to spin around during its trip around the Moon, just like the Sufi dancers do. Just to make things more difficult, he had his robot turning in the opposite way (anti-clockwise looking from above) to the turns taken by virtually every other student in the class (clockwise looking from above) – and, to his credit, he succeeded.

The video below shows that his idea, which he applied to a LEGO EV3 robot, can also be achieved by our Fusion robot. Since this student was a boy, and boys very often want to use a speed of 100% for everything, this video also demonstrates what happens to our Fusion robot when it spins at a full motor speed of 100%. I'm not sure the student realized it, but this made his task much more difficult. Using higher speed "stops" and "starts" very often result in much wheelspin. If the floor is even the tiniest bit uneven, the use of high speeds resulting in wheelspin will make the robot's final direction much less reliably predictable. This problem of high-speed wheelspin is the reason we suggest that most of the students run their robots at a speed of 40% to 50% of full power during all testing runs, there being much less chance of wheelspin at these lower speeds. They can try higher speeds later when they are preparing for competition events!

## ***Fusion goes around the Moon - the Sufi Way!*** FB6c

The video below shows Fusion's version of a "Sufi" extension to the "Around the Moon" Challenge.

Video - Web - USB

### *Fusion Loops Around the Moon.* FB6d

A second extension to the "Moon Challenge" is for the student to teach their robot to go in a loop around the "Moon", before returning to "Earth".

At one of the Adult Education parent/child weekend robotics workshops run by Yaya Lu and myself, one of the girls wanted to program her robot to do a loop around the Moon before returning back to Earth.

This Fusion program is a little different from the previous "Blockly Moon Challenge" programs. The "Rotate" command that we previously used to change Fusion's direction, is not appropriate in this case. The previously used "Rotate" command spun our Fusion robot around on the spot. To go in a loop around the Moon, we will need to use separate motor commands for each motor. In MRI's Fusion implementation of Blockly, separate motor commands do not have a time limit built in. You will see that we have to add a separate "Time" command limit *after* the two motor commands. The motors will continue running until the "Time" command is finished. This is discussed in the video.

The video below demonstrates that a Fusion robot is capable of imitating this girl's robot solution to the Moon Challenge.

Video - Web - USB

### *Other "Moon Challenge" extensions for Fusion to Try?*

The Moon Challenge seems to have acted as a stimulus to student imagination. I particularly remember a boy student was extremely enthusiastic about his "super-dooper secret spy robot". It left Earth in the opposite direction to everyone else's robots, heading away from the Moon. His reasoning was that anyone on Earth would think his robot would be going to Mars or Venus. What his robot was actually going to do, he said, was to go around the Moon and spy on a secret enemy base on the far side of the Moon, an enemy base which we could not see from Earth. He knew that people on Earth only ever see about one half of the Moon, and that no one on Earth can see anything on the far side of the Moon when they are standing on the Earth. This is not widely known, but he knew it! He then taught his Robot to return past the Earth, turn around in Space, and then come back to the Earth from his Mars or Venus direction, so that no-one on Earth would know where he had been spying! He was very proud of his super-dooper secret spy robot, and he completed his version of this Challenge quite satisfactorily – to the applause of the other students…

How many ways can you teach your Fusion Robot to "Go around the Moon"? Perhaps inspiration can be gained from these old videos of attempts using different robots. The runs vary from excellent to disastrous 😉:

Video - Web - USB
Video - Web - USB

### *Downloadable Arenas:*

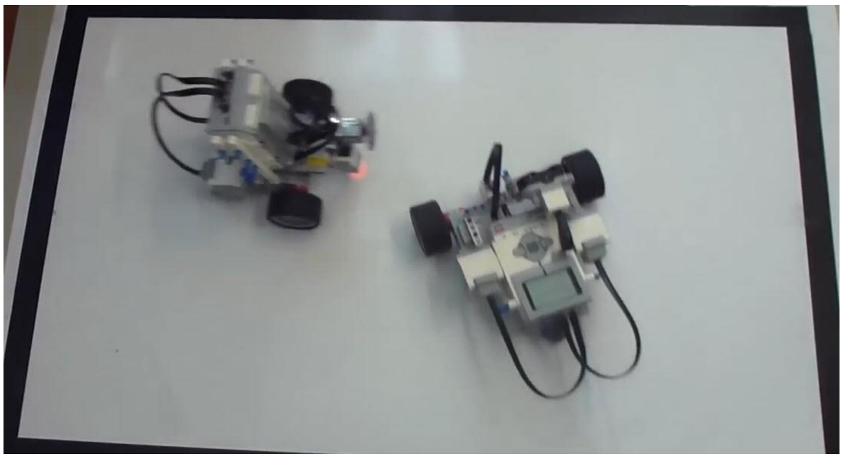
If you have access to an A3 printer, an arena that we have used for this tutorial is available for download Web – USB. Our original arena is here Web - USB. You could also reuse the arena that we have already used in the "Alien" tutorial on page 10 of this publication. A pdf version of this arena is available for download Web – USB.

# Robot SUMO using Fusion FB7

Japan has a sport called SUMO in which each person attempts to push the other out of an arena.

We would not have much chance of competing! But we can have fun with a robot version of SUMO.

To enable our robot to play SUMO, we need to make our Fusion Robot able to sense the edge of the SUMO arena. The downloadable (Web – USB) arena we used has a white center with black borders, as shown below. To detect the black line that surrounds the arena, we need to find out how to use Fusion's Optical Distance sensor.



## *How to use Fusion's Optical Distance Sensor.* FB7a

To play SUMO, Fusion needs to be able to tell when it is somewhere inside the arena, and when it has found the thick black line surrounding the SUMO rink. We will use the Optical Distance Sensor to let Fusion know which part of the rink it is in.

This video demonstrates the use of Fusion's Optical Distance Sensor to detect whether the Sensor (and the hence the robot) is in the white center of the SUMO rink, or if it has found the black edge of that rink. The difference in sensor readings obtained from the Optical Distance Sensor when it is over the white area, and when it is over the black area of the rink, can later be used to teach Fusion how to stay inside the SUMO rink.

Video - Web - USB

***Update:*** In the latest version of the Fusion Blockly software, the "Example" program used in this video has been moved to the menu location:
Help/Examples/Intermediate/OpticalDistanceSensor/odsRead.blk

## *Keeping Fusion inside the SUMO Rink* FB7b

Video - Web - USB

Th video above is a "first look" at teaching our Fusion Robot to stay inside the SUMO rink.

***Note:*** Viewers with keen eyes will note some changes, such as different "Run" and "Stop" symbols in the Blockly editing screens of these two videos. You are using an updated version of Fusion's operating system that was not released when I filmed these videos. I hope to update these videos when I have completed the "Fusion Python" course that is a companion to this course...

## *Extra: Keeping Fusion inside a Colored Rink* FB7c

Video - Web - USB

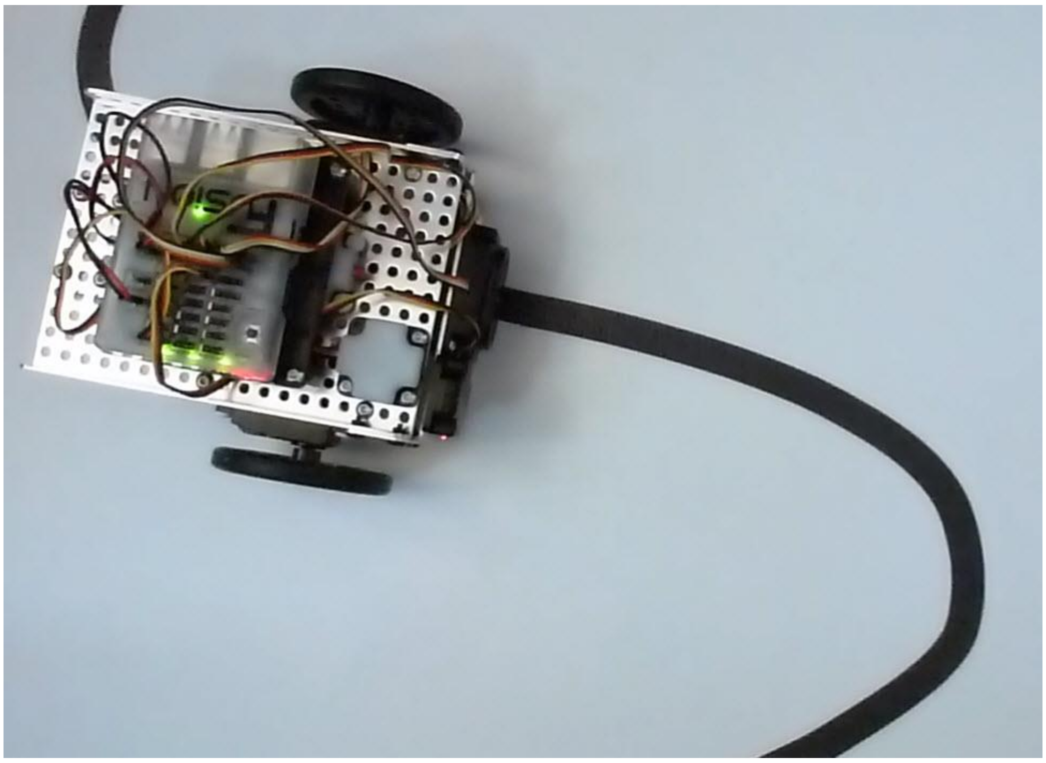
There are occasions when the rink is all one color, with no surrounding one-color rim. Fusion can handle this as well, but needs a Color Sensor. This sensor is not part of the Fusion Base Kit, but is available for separate purchase from Modern Robotics. When the Color Sensor is used, you will need to know the "Utility frequency" of the electricity supply in your country. In Tasmania this is 50 HZ, in the USA this is 60 Hz, and you can find the Utility frequency for your country by clicking here.

***Note:*** Currently (*November 2018*) only one Color Sensor may be connected to the Fusion Controller at a time. My understanding is that this is a limitation of the current Fusion implementation of Blockly, not a limitation of the *Color Sensor/Core Controller* combination itself. Thus, this limitation of only using one Color Sensor at a time could change in a future implementation of Blockly.

***Further Technical Information:***

You do not need more information about these sensors to complete this tutorial. However, if you are a technician who is interested in further information about these sensors you can find by more by clicking on Optical Distance Sensor, and Color Sensor.

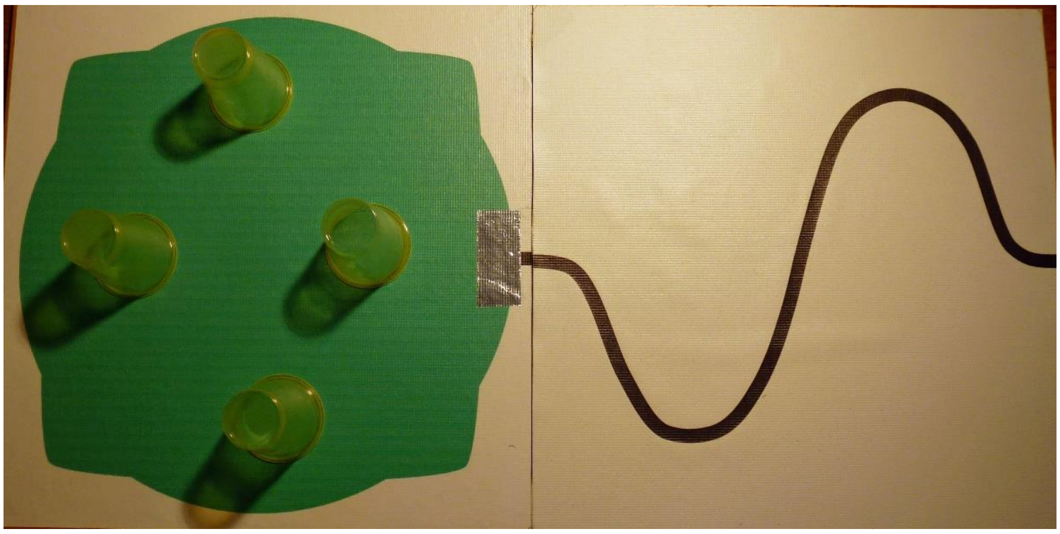
## Fusion follows a line edge FB8



This is a "first look - proof of concept" video that shows Fusion can follow the edge of a line. Please note the slot "A0" to "A1" correction in the video. When Fusion is shown following the edge of a line, it is using the corrected program.

Video - Web - USB

## Fusion's Swimming Pool Challenge [FB9]



Can our Fusion robot clear the toys from the green swimming pool?

We imagine that our class has gone on a bus trip to a beautiful swimming pool in the wilderness. They have only been playing for a little time when there is a sudden heavy shower of rain. The driver of the bus is concerned that the shallow stream the bus crossed may suddenly get deeper, so he hurries all the children into the bus and they leave quickly before the stream can rise. Unfortunately, this leaves the teacher to clear, from the swimming pool, all the toys that the class were playing with. Unfortunately, the teacher can not swim. So, we need to program our special amphibious Fusion robot to follow a winding trail through the wilderness, to next detect the silvery white sands of a beach on the edge of the pool, and to then push all of the toys to the edge of the pool so that teacher can collect them without getting wet.

How can we teach our Fusion robot to do that?

### *Fusion's "Clear Pool" Function* FB9b

This video is about modifying our previous SUMO program, to assist our Fusion Robot to "Clear the Pool of Toys".

We take the main working part of our SUMO program. and move it into a "Blockly Function" body. This gives us an extra "Clear Pool" Blockly block, which we insert into our main program in the place from which we have just removed our code. Since the SUMO ring had a white center, and a black surround, and our swimming pool has a dark (green) center and a white surround, we need to make a couple of minor changes to our new "Clear Pool" function (see video).

Video - Web - USB

However, as you can see from the video, it all works out quite nicely!

## *Fusion’s Edge-Follow Function* FB9c

Video - Web - USB

This video demonstrates how to change the Edge-Follow Blockly program we previously used, into an Edge-Follow Blockly Function. The use of the Edge-Follow Function will simplify the program we will next develop, with the aim of teaching our Fusion robot how to clear the toys from the swimming pool in the wilderness.

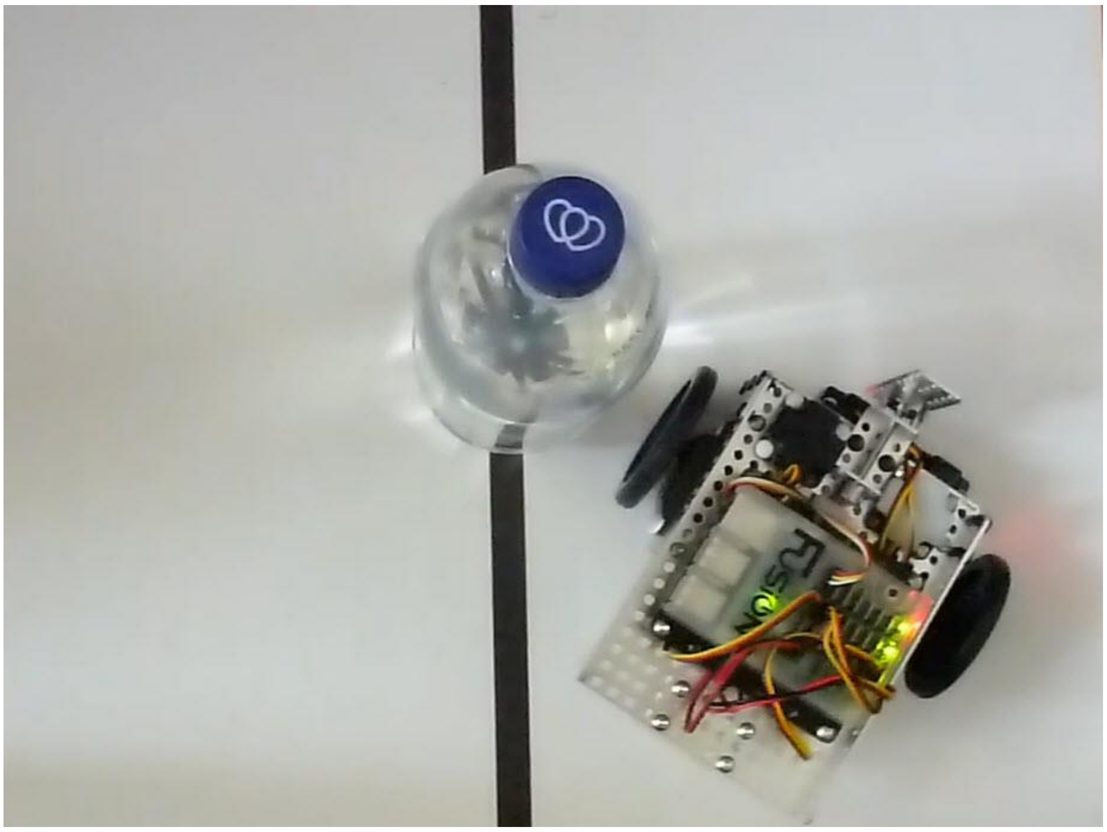
## *Fusion attempts to “Clear the Swimming Pool”* FB9d

In this video we combine the Edge-Follow and SUMO Blockly Functions in a program that will teach Fusion to follow a curving path through a pretend wilderness, cross a "silver sand beach", and clear left-behind toys from a swimming pool by pushing them to the edge of the pool so they can be collected by our non-swimming teacher. It does this beautifully!

Video - Web - USB

This Challenge illustrates nicely the major program simplification that can be obtained by the use of functions. *(**Note:** My half-a-century background in lots of other languages has meant that I have been used to calling these functions "procedures", so if you hear the word "procedure" in this video, I mean "function". Sorry about that.)*

## Can Fusion Go Around an Obstacle? FB10



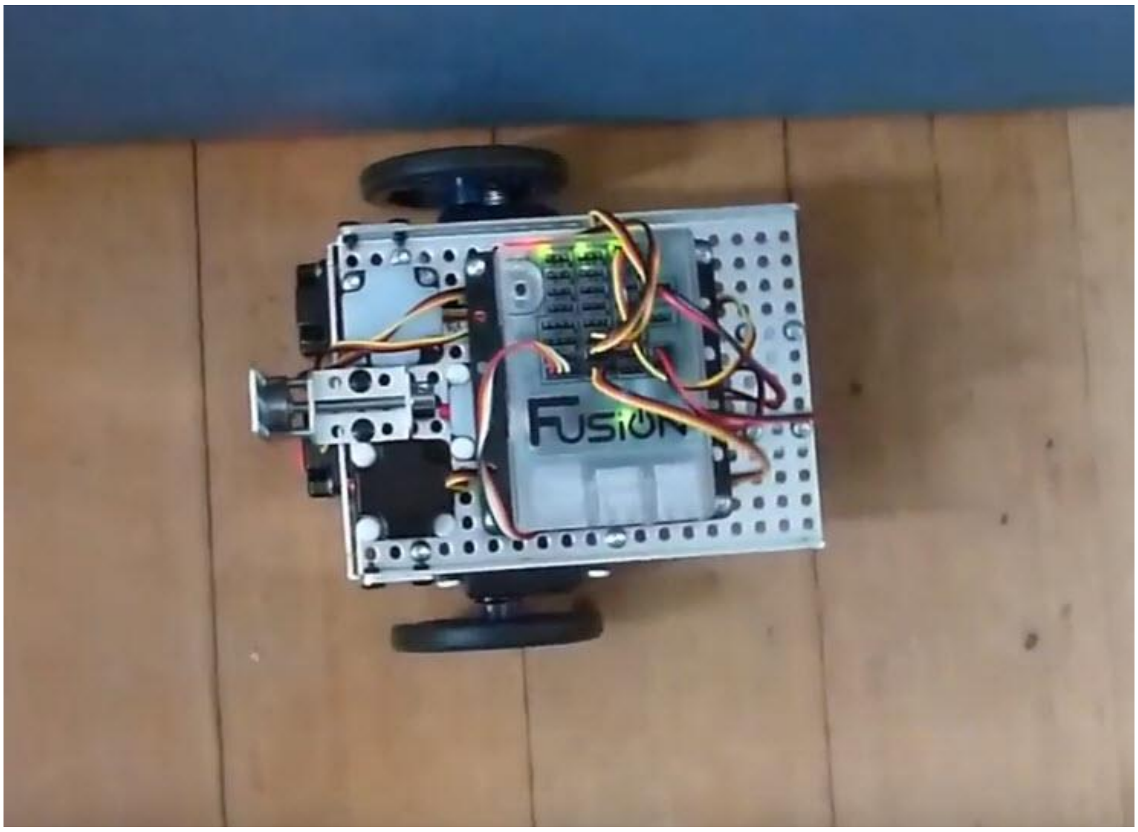
Video - Web - USB

This is a "proof of concept" video. It demonstrates that Fusion can be programmed in Blockly to avoid an obstacle on a line, and regain the line to continue edge-following. Works nicely...

***Further Technical Information:***

You do not need more information about these sensors to complete this tutorial. However, if you are a technician who is interested in further information about these sensors you can find by more by clicking on Optical Distance Sensor, and Touch Sensor

# Fusion Follows a Wall FB11



In this video we see if we can get our Fusion robot to follow a wall.

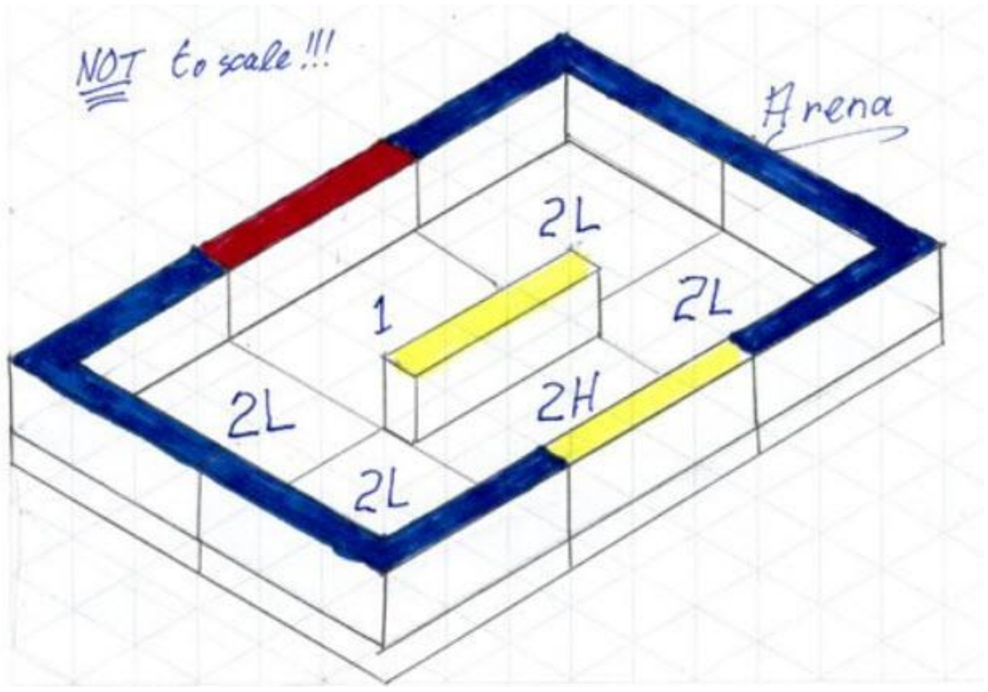
Video - Web - USB

This is another "proof-of-concept" video. We wanted to see how well we could get Fusion to follow the wall using a wall-follow function that is essentially the same in logic format as the line-follow function that we used earlier with Fusion.

As can be seen in the video, we have made some small changes to the program. However, the actual logic of the "if-do" and the "motor arrangements" are still essentially the same.

This similarity opens up the possibility that, if the Fusion tutorials that could be eventually written were carefully planned, basically almost the same function could be used for a first look at both "line following" and "wall following". This could result in a considerable simplification of the tutorials, which would be valuable if they were to be presented into the Primary School grades.

# Fusion and the Hippodrome Challenge FB12



In this "proof of concept" video, we attempt to teach Fusion to go around a hippodrome without failure.

Video - Web - USB

Trying to teach your robot to go around a Hippodrome, or Coliseum, was always an exciting thing for some of our younger classes. Even the first attempts, although sometimes somewhat shaky, were always exciting to the classes. For example, click here to see these Grade 5 classes encourage their LEGO robots as they attempt to go one lap around their arenas.

This is one of the few robot-to-robot competitions we used. We always tried to emphasize the fun, rather than who was the winner.

We went through several variations of the hippodrome. A typical early version can be seen by clicking here.

The main problem was these versions were difficult to fit in a car to transport between schools. A secondary problem was that some of these variants were too heavy for primary school students to help by carrying them in from the car to the school.

Around 2008 we devised the maze system which you can see being used in the video at the start of this tutorial. With the exception of the yellow-tipped maze element, all the other maze elements are individual, stack-able, easily carried in the car between schools, and light enough for primary school students to be able to get a sense of achievement by helping to carry them from the car to the school. Once in the school, the individual pieces can easily be arranged into a maze. For example, here is a video which shows how the Coliseum, or Hippodrome, is put together:

Video - Web - USB

The other nice thing about this system is that these individual elements can be easily rearranged so that the maze is not the same each time the students try to teach their robots to go around it. The webpage you can see by clicking here shows how a selection of maze elements can be rearranged in a whole variety of different maze shapes.

The individual stack-able maze elements are sturdy. The ones in the video above are about nine years old. Even though they have been used in many schools, and are showing their age, they still provide a good challenge.

The individual maze elements are also quite easily constructed by an amateur. In 2011, when Yaya Lu did not have enough maze elements to make the type of maze she wanted, she made some more herself. She was in Grade 7 at the time. You can see a video of her making one of these at the top of this web page (click on the web cam images to see the videos):

These maze elements are well worth thinking about. Lots of educational fun for your students!

## *Extra:*
## *Fusion , the Hippodrome, and 2 Optical Distance Sensors* FB12b

Video - Web - USB

In this "proof of concept" video, Fusion goes around the Hippodrome using two Optical Distance Sensors. The Fusion Basic Kit includes one Optical Distance Sensor, but others can be obtained here. If you have converted a Spartan Robot by replacing the Core Spartan Controller by a Fusion Controller (see page 29), you will already have two Optical Distance Sensors that were supplied as part of the Spartan Kit.

This video is very similar to the previous video, with only small changes being necessary. The replacement of the Touch Sensor by a second Optical Distance Sensor is one change needed, leading to small changes in the main program, and in the function that turns Fusion 90 degrees at the end of the hippodrome.

***Note:*** The small light that can be seen pointing downwards during Fusion's run is from a third Optical Distance Sensor that is not used during this run.

### *Extra: Fusion , the Hippodrome, and the Range Sensor* FB12c

In this video we look at the use of a Range Sensor, combining with an Optical Distance Sensor, to assist Fusion to navigate around a Hippodrome. The Range Sensor is unusual in that it combines two sensors, an Optical Distance Sensor, and an Ultrasonic Sensor.

- The Ultrasonic Sensor is good for detecting objects at a greater distance, but has problems detecting objects that are close to the Robot.

- The Optical Distance Sensor is good for detecting objects that are close to the Robot, but not so good for objects that are further away.

The two sensors are combined in the Range Sensor to provide a good object-recognition sensor that can be used to detect objects both far and near.

This "proof of concept" video demonstrates how the Range Sensor's Optical component can combine with a separate Optical Distance Sensor, to allow Fusion to navigate around a Hippodrome.

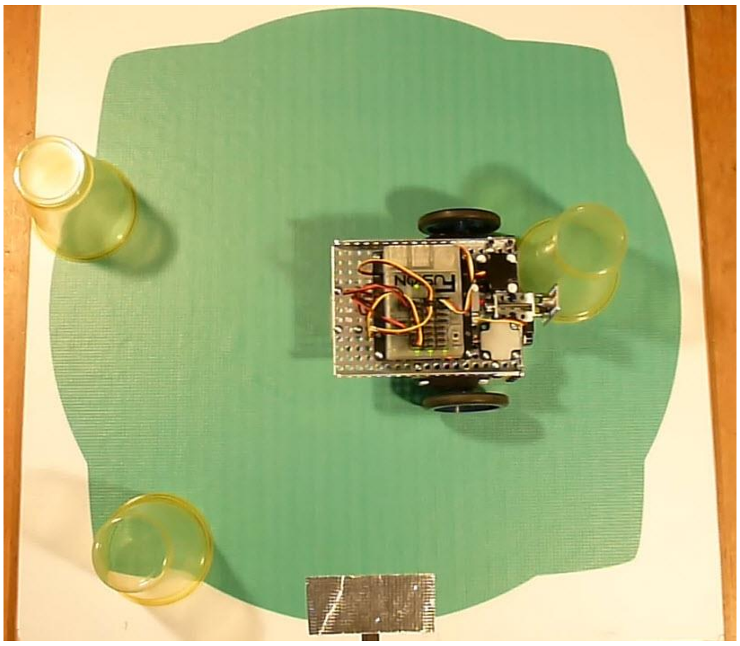
Video - Web - USB

**Note:** Neither the Fusion nor the Spartan Basic Kits include a Range Sensor, which can be obtained here.

***Further Technical Information:***

You do not need more information about these sensors to complete this tutorial. However, if you are a technician who is interested in further information about these sensors you can find by more by clicking on Optical Distance Sensor, and Range Sensor

# Extra: Using Fusion's Range Sensor to Clear Toys From the Swimming Pool FB13



This video explores the use of the Ultrasonic portion of the Range Sensor to clear the Swimming Pool more quickly than the previous method we used.

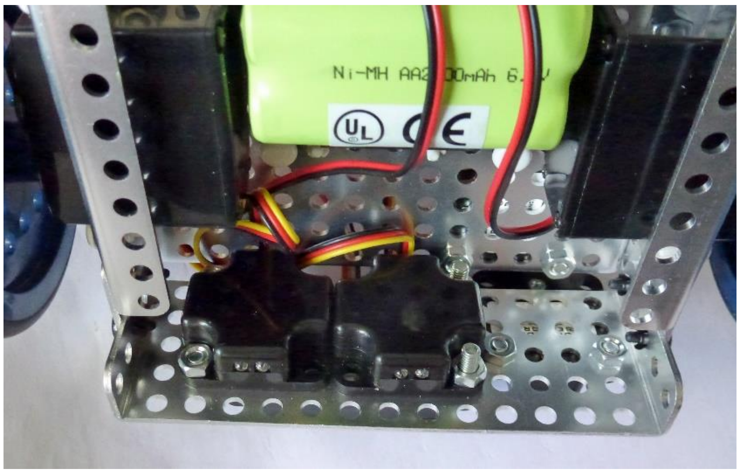
When we cleared the Swimming Pool previously, we used a fixed turn, hoping to eventually push out any toys. However, using the Ultrasonic portion of the Range Sensor, we can actively search for the toys, find them, and push them out directly. Although we have not done so in this example, we could also potentially check that the toys have all been pushed out, by checking that Fusion has made one rotation without pushing out any toys.

Video - Web - USB

***Note 1:*** Neither the Fusion nor the Spartan Basic Kits include a Range Sensor, which can be obtained here.

***Note 2:*** There are competition circumstances that would make it advantageous to mount more than one Range Sensor on a Fusion robot. However currently only one Range Sensor may be connected to the Fusion Controller at a time. My understanding is that this is a limitation of the Fusion implementation of Blockly, not a limitation of the Range Sensor/Core Controller combination itself. Thus, this limitation of only using one Range sensor at a time could change in a future implementation of Blockly.

# Extra: Fusion Follows a Line Using Two Optical Distance Sensors FB14



In this "proof of concept" video, we take a first look at Fusion following a line using one Optical Distance Sensor on each side of a line. In our experience, this is an option preferred by many students to the one-sensor edge-following option. The Fusion Basic Kit includes one Optical Distance sensor, but others can be obtained here.

## ***First Method:*** FB14a

Video - Web - USB

This is another brief "Proof of Concept" video that shows one method of teaching Fusion to follow a line using two Optical Distance Sensors.

This method re-uses the "Edge Follow" functions used in a previous video in this series, but with small alterations so that each sensor follows a separate edge of the line.

## ***Second Method:*** FB14b

Video - Web - USB

This brief overview "proof of concept" video is a follow-up to the previous "Line Follow" video. It uses an identical robot to the previous video to demonstrate an alternative strategy to achieve line following.

# Starting and Stopping Fusion During a Run FB15

Many competitions, such as the Tasmanian RoboCup event, demand that the robots operate autonomously. Effectively, this means that the student competitor must be able to stop and start their robot even when their laptop is turned off.

This would seem to be a bit of a problem in the case of the Fusion robot, because the program is started from the laptop, and may appear to an official in a competition to be controlled by the laptop even during a run. In the current (November 2018) implementation of Blockly for Fusion, there does not appear to be a provision for a "remote" start and stop of the Fusion robot.

However, we can simulate that facility by the use of the touch sensor, as shown in the video below. The program can be started from the laptop, and the laptop then turned off. The Fusion robot will then do nothing until the Touch Sensor is pressed. The program will run until the program terminates itself, or the student presses the touch sensor again. You can see this working in the video below.

However, this solution is not foolproof. It depends on the student's Blockly function being fairly short, so that the robot can check reasonably frequently if the touch sensor has been pressed. If the program gets into an infinite loop inside the student's Blockly function, pressing the Touch Sensor will not make any difference because the student program does not end, and there is no opportunity for Fusion to check whether or not the touch sensor has been pressed.

It would be preferable for there to be some sort of facility within Blockly that would permit the student to start and stop a Blockly program during the competition runs. However, in the current absence of such a facility, the approach shown above will have to do, until we can find a better way of providing this "local stop and start" facility on the Robot.

Video - Web - USB

# *Technical Extras:* FBTe

## *Fusion Robot – Introduction* FBTe1

This page contains a summary of semi-technical information that may be of interest to Teachers or Mentors. If you are a beginner who is eager to get on with building your Fusion Robot, ignore the stuff below, and go to page 3 and start building your Robot.

Fusion is a Robot kit produced by Modern Robotics Inc.. It is controlled by a Raspberry Pi 3 mini-computer, combined with a Modern Robotics circuit board that allows the Raspberry Pi to control all of Modern Robotics' Fusion motors and sensors. These are protected by a transparent enclosure, with slots on the top for connections to sensors and motors.

You can tell your Fusion robot what to do using a version of Google's Blockly computer language, which is an excellent language for beginners. For more advanced users, the computer language Python (which is very popular in industry) can be used with Fusion. If you have not used Robots before, we strongly suggest you start with Blockly, because Blockly is easier for beginners, and Fusion is designed to make any later conversion from Blockly to Python a reasonably straight-forward process.

Fusion is unusual in that the documentation and compilers are all mounted within the Fusion Robot itself. Because Fusion can act as a local Wi-Fi source, all that is needed for you to control your Fusion Robot is a Wi-Fi link from your laptop to your Fusion Robot. Once you have connected these two using a browser (preferably Google's Chrome browser) you do not need a connection to the Internet. Not needing the Internet can be a blessing if your Internet connection is as slow as a tortoise with sore legs, or if your school has forbidden student laptops to be connected to the internet. In both cases your local Wi-Fi link between your laptop and your Fusion robot is all that is needed to get started.

All this software is designed to work within a browser. Modern Robotics recommends using Google's widely-available Chrome browser, versions of which are available on most recent computers running Microsoft's Windows operating system, recent Macintosh computers, and computers running many Linux distributions. If your computing device is using an older or different operating system, check Google's advice here.

## *Updating Fusion's Operating System* FBTe2

Periodically, Modern Robotics makes available an update to Fusion's Operating System. They have just (November 2018) made available v1.0.5, an update from the previous v1.0.0. This adds the ability for Fusion to participate in "Data Logging", where readings from a sensor can be recorded over a period of time.  This can be excellent for school projects. An example of a report produced by the best of my previous mentees who used "Data Logging" can be seen here.

Guidance on how to update your Fusion's Operating System can be viewed in the web here, or via a video Web – USB.

## *Fusion Controller Reference* FBTe3

Modern Robotics has recently provided additional technical documentation for the Fusion Controller. It can be referenced by clicking here.

## *Fusion Sensor Reference* FBTe4

Modern Robotics has recently provided additional technical documentation for Fusion's Sensors. It can be referenced by clicking here.

## *Fitting a Fusion Controller to a Spartan Robot* FBTe5

Modern Robotics sells kits for both Spartan and Fusion Robots. Apart from the Controllers, the physical components in both robot kits have many similarities. If you own a Spartan Robot, and decide you want to control your robot using Fusion's Python computer language instead of Spartan's C computer language, it is possible to successfully mount a Fusion Controller on a Spartan robot base. The following notes comment on what I did to achieve this change.

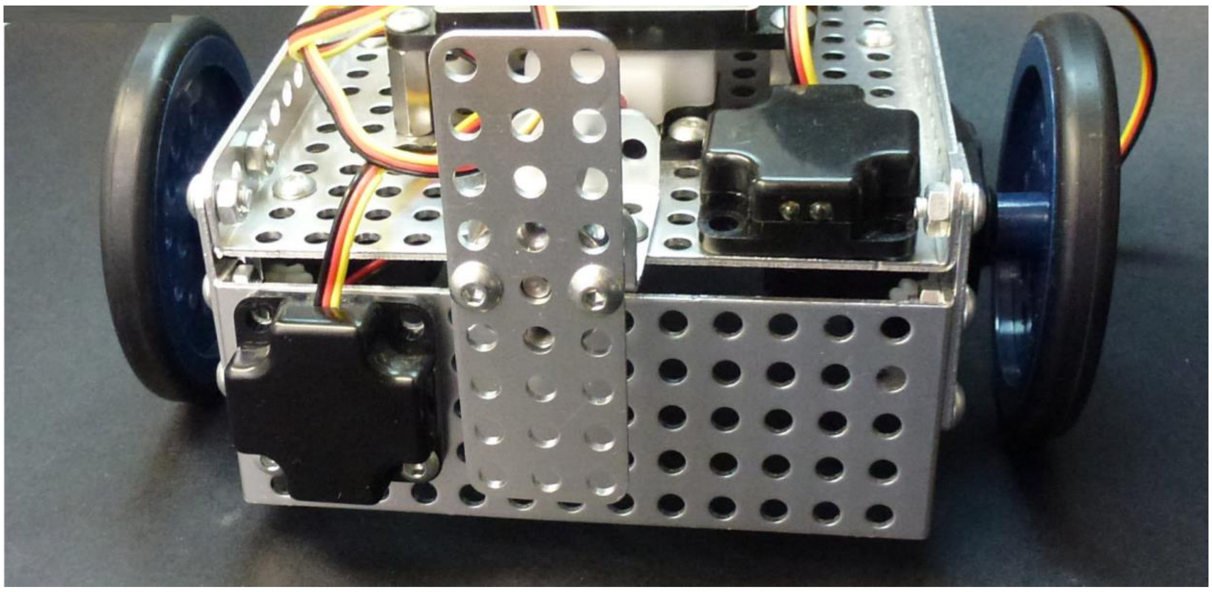
Since the metal components of the Fusion Base Kit are very similar to the metal components in Modern Robotics Inc.'s Spartan base kit, replacing the Spartan controller on my existing Spartan Robot by a Fusion Controller was fairly straight-forward. My Fusion build mostly followed the Spartan build you can find in Modern Robotic's Spartan build pdf, and in this YouTube Video. However, some changes were necessary.

The Fusion Controller is bigger than Spartan's Arduino controller. Hence, you should mount the Fusion Controller in the position shown in Step 3 and "Step 14" (the second "Step 12") of Modern Robotics' Fusion Build Instructions that are available here. The Fusion controller is mounted sideways to allow sufficient space to enable sensors to be mounted behind the controller, if this is required in the future.

The sensors supplied with the Fusion Base Kit are different from the sensors supplied as part of the Spartan Kit.

The Fusion kit includes one Optical Distance Sensor and one Integrating Gyro Sensor.

The Spartan Kit includes two Optical Distance sensors. I found it convenient to change the standard Spartan mounting position of the second (left) Optical Distance Sensor, to that shown in the image below. This change will help prepare your Robot for the "Robot SUMO Using Fusion" (page 14) and "Fusion Follows a Line Edge" (page 17) tutorials.



Enjoy! 😊

# Where to go next? FB17

It has been fun using Blockly to teach our Fusion robot how to do what we desire it to do. What comes next?

My first impression is that Blockly seems quite suitable for lower middle school use and above. It is also worth noting that it is possible to go much further with Blockly than we have demonstrated in these "Absolute Beginner" tutorials, but that is for another day…

The other really nice thing about Fusion, is the included access to Python as a programming language. We have turned Python off in these Blockly-based "Absolute Beginner" lessons, to save distracting students. However, my impression is that a fairly clean transition could be made from Blockly to Python - and the libraries available for use in Python are fabulous! This could probably make Python with Fusion (and its extensions) suitable for use in upper middle schools right through to University.

We are currently (November 2018) working on a www.DrGrae.me intermediate-level course demonstrating how to transition from Blockly to Python, and hope to have it available for use with Fusion by early 2019. 😊