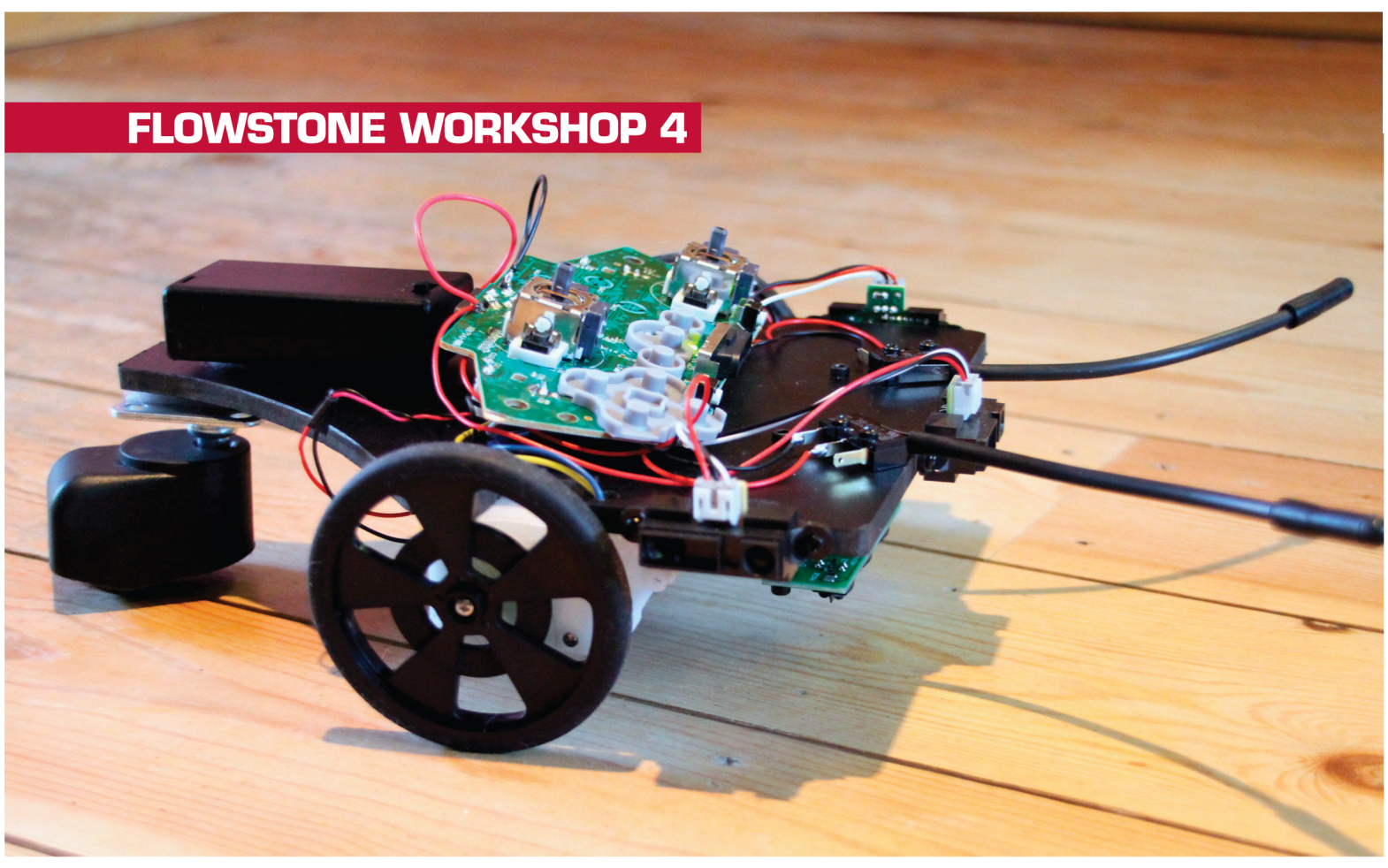
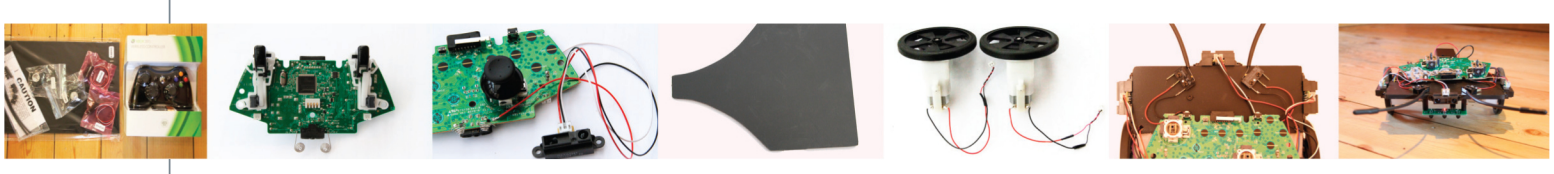


by Carl Owen

Behold The xBugBot

A hacked, wireless, autonomous xBox controller!



Welcome to the fourth of our FlowStone Workshops where we give a beginners guide to programming Robotics using the FlowStone FREE programming language. For this workshop we thought we would do something a bit different: Behold - xBugBot! What is it? It's a fully autonomous wireless robot made out of a hacked xBox controller!

FlowStone, the graphical computer programming language, supports various games controllers, PSII, WII Mote, xBox and more. It was when I was programming a simple computer game in FlowStone, called Klingon Killer (available as a download on the DSPRobotics Forum), to test out how well the xBox controller worked, that the idea came to me. When you think about it, the xBox Controller is a really cheap USB DAQ (data aquisition) board. It has several analogue inputs (used for the joy sticks), several digital inputs (buttons etc.) and two PWM motor controllers (used for the rumble vibrators).

Some of the xBugBot parts.

The xBox Wireless Receiver.

I was using the wired usb xBox controller, and it was looking at the real-time data inside FlowStone and playing with the rumble options on the controller that gave me the inspiration. I wonder if you could hack this to make an autonomous robot? Even better if it was wireless! So wasting no time we ordered a wireless xBox controller and wireless USB receiver (you need this to connect it to a PC), which landed on my desk the next morning!

A few hours later xBugBot was born and autonomously navigating round the kitchen!

KIT PARTS

1 x FlowStone Software (FREE)
1 x Wireless xBox controller
1 x xBox USB wireless receiver
2 x Small motors and gearboxes e.g., Solarbotics Motor 8
2 x Wheels e.g., Solarbotics 2- 5/8" wheels
3 x IR range sensors e.g., Sharp GP2D12
3 x line sensing kit or line detection sensors and resistors e.g., TCRT5000
2 x Micro switches & whiskers e.g., LynxMotion Bump Switch Kit
1 x Battery holder with switch e.g., 2xAA Battery Holder with

Switch
1 x Castor wheel. Eg ½" ball Caster wheel
1 x 6mm Sintra board
A few bits, cable ties, tape, sticky Velcro, solder, screws etc.

IDEAL FOR EDUCATION

The more you think about this little project the more it makes sense for education, compared to some of the off the shelf kits you can buy at great expense, this is great value for your money at around $99!

EDUCATIONAL BENEFITS

It's a Hack - kids love the fact that it is something they can identify with, like a game controller that they probably have at home. How cool is it to hack it into something else. It's Autonomous – Let's be honest if it wasn't it would be a remote controlled car with little educational benefit. The fact that it's programmable means that different teams can work on their own algorithms to make the best roving xBugBot possible. It could even be entered into some national robot competitions. If there's anyone out there that wants to host the xBugBot world games, let me know!It can follow Lines – using the line following sensors it can also follow lines on the ground. So it can have two modes, fully autonomous or line follower or both.

It's wireless with telemetry – Now this is something even the most expensive robot kits rarely have. Using telemetry, similar to Formula 1 race car technology, you can see the live data from the robot sensors in real time and even modify the code on the fly. It uses FlowStone Free – This is the graphical programming language teachers can easily understand with little effort. Kids can visualize the programming methods much faster than with a conventional language like C++. What's even better is that there is no compiling necessary, this might seem like a little thing but it has a huge impact in the classroom. FlowStone runs in real time – All of the time! So when you develop your project the results are instantly displayed as you type. This means that errors and bugs are caught as you make them saving literally hours of time when programming. Contrast this with the old procedure: type, compile, test and try again. It's fun – this is a fun robot that is just great to play with, not too big, not too small, cheap to run, and easy to program! Interested?

THE ROBOT BUILD

There is a full 18-page instructional PDF on the DSPRobotics website: www.dsprobotics.com/Files/XBugBot.pdf. Here are the basic skills required:

- Cutting the Sintra board (we used a jig saw)
- Soldering the wires from the sensors on the xBox PCB
- Drilling a few mounting holes (a Dremel comes in handy)
- Creative use of sticky Velcro and cable ties!

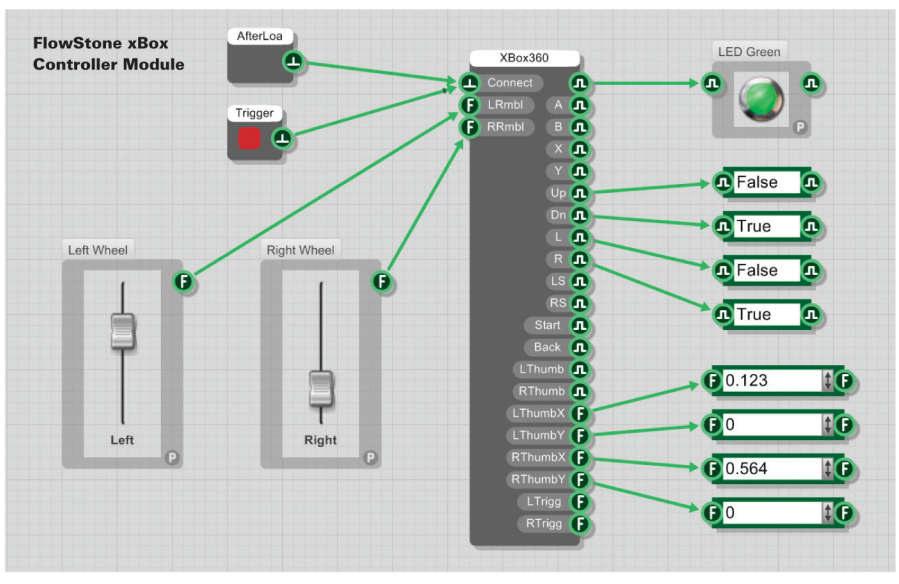
PC SETUP

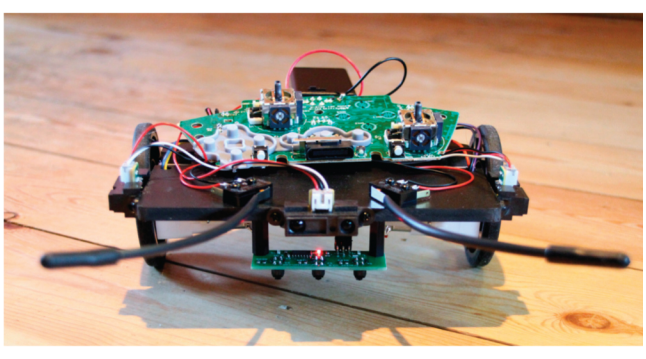
Setting up your xBugBot on the PC couldn't be easier, it works on XP upwards, and the xBox drivers are already included in Windows. So it's Plug and play when you connect the wireless xBox usb receiver, it detects the device and automatically installs the driver.

Next is to pair the device, as the xBox receiver can talk to up to four controllers. To do this press the button on the xBox receiver then the small button on the controller board (refer to the xBugBot manual for more details). Finally you need to install the FlowStone Free software which is downloadable from the DSPRobotics site.

PROGRAMMING

Open FlowStone and search for the 'xBox 360' module, drag this into the schematic window. In order to connect it to the xBox hardware installed you need to start the module by sending a Trigger to the Connect input. This can be done automatically by using an 'After Load' module or manually using a 'Trigger Button' Module. That's it, you should now be connected to the wireless hardware, you can check this by adding an LED module to the 'Connected' output on the module and a few 'Float' and 'Bool' display boxes connected to the





outputs. When you press any of the xBox controller buttons or activate the sensors you will see the real time readouts in FlowStone.

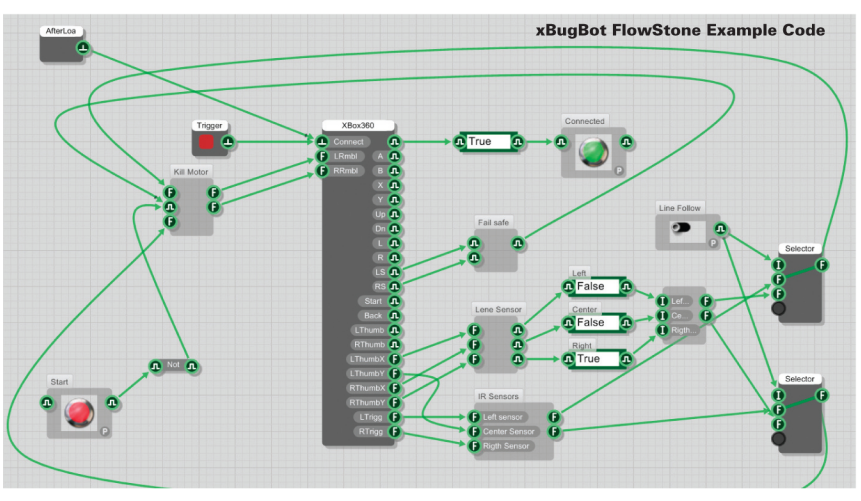
To test the motors you can add a couple of 'Sliders' to the rumble inputs and control each motor independently.

AUTONOMOUS ROVER

After you have tested your xBugBot hardware you can start the fun stuff of programming it to become an autonomous rover. Since you have three IR distance sensors you can detect objects in front and to the sides of the robot, so using these you can make decisions on how to avoid obstacles. For example if the robot detects an object in front and to the right then turn left etc. We have made a simple FlowStone program example you can download and test that does just this: it detects objects and attempts to steer around them. We have also used the front bump sensors as insurance so if the robot gets too close to an object the bump sensors are triggered and the robot is shut down.In the example program we have kept things simple by employing a simple proportional response to objects, if you wanted to go to town you could use FlowStone's PID module for a professional level app.

LINE FOLLOWER

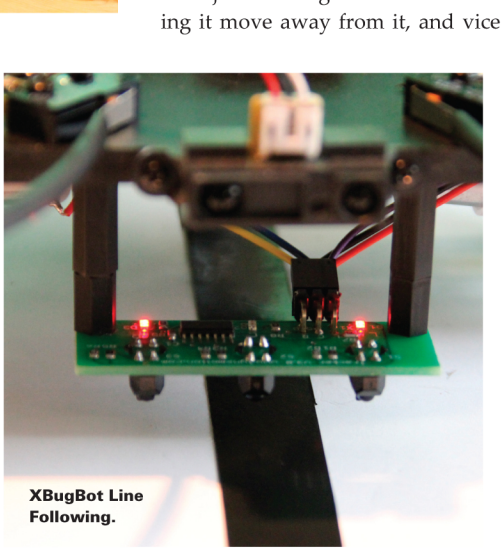
Using the three line follower sensors the xBugBot can detect and follow lines on the floor. This can be a mode that you switch on or off in the software (live) or it could be an automatic thing, where if no line is detected the robot is in Autonomous mode or if a line is detected, follow it! We used some inexpensive black insulation tape on the ground that worked really well.



EXAMPLE TEST SOFTWARE

Here are the basics of the FlowStone example project. Once the software loads it automatically connects to the xBox hardware installed on your PC using an After Load Module. To start the robot you need to click on the Start LED. This goes via a Kill Motor module that reads the bump switches and stops the motors if it hits anything. The IR distance sensors are processing in the IR sensor module where the signals are first averaged to reduce the noise and then multiplied by a gain factor on a slider to get the correct levels.

For the motors the program is very basic. If the xBugBot senses an object to its right it slows down the opposite wheel a little helping it move away from it, and vice versa. This works remarkably well, so the xBugBot will actually follow a wall! For the front sensor there is a trigger level also set by a slider (distance to front object) that once triggered requests that the robot turn for 3 seconds (roughly 90 degrees). This is done by stopping one wheel and sending full power to the opposite wheel. Sounds simple but it seems to work really well, obviously this is very basic but you could add some advanced maze solving code if you wanted. Using FlowStone V2 you can now also program in 'Ruby' code directly inside FlowStone making it possible to actually code your decision engine if you preferred.



XBugBot Line Following.

The line following in the example is turned on with the Line Follower Switch, and simply keeps going straight while only the center line sensor is detected, but once one of the left or right sensors goes over the tape the opposite wheel is slowed down to bring it back in line. Again simple but effective!

CONCLUSION

The xBugBot is a great fun project that is only limited by your imagination and your programming skills. This is ideal for education and at around $99 is fantastic value for the money. Add to this the full power of a professional programming language like FlowStone and there are no limitations to what you can do. It would also be possible to pimp your xBugBot by adding a wireless webcam and using the powerful image processing of FlowStone and the PC. This could be suitable not only for children but also for a university level project. ◎

Links
DSPRobotics (free Flowstone), www.dsprobotics.com
Robot Build Instructional PDF, www.dsprobotics.com/Files/XBugBot.pdf
xBugVideo, www.youtube.com/watch?v=1xGA5bDD-Ug

For more information, please see our source guide on page 81.

See The Video!
Scan bar code or type in find.botmag.com/051204