HUMANOID BUYERS GUIDE

ARDUINO CONTROLLERS EXPLAINED

ROBOT

THE LATEST IN HOBBY, SCIENCE AND CONSUMER ROBOTICS

FLOWSTONE
THE NEW GRAPHICAL PROGRAMMING LANGUAGE

SEE THE VIDEO!

Scan this code on your smartphone with a bar reader app or type in find.botmag.com/031106.

BALLOONBOT GRIPPER

FACE TRACKING BOT

BUDGET BALANCING BOT

VIRTUAL ROBOTICS GAMING WORLD Pg. 78

BOT BATTERY TESTING SECRETS

WWW.BOTMAG.COM
MAY/JUNE 2011
$6.99
06
0 09281 04748 3

ROBOT MAY/JUNE 2011 ISSUE 28 BOTMAG.COM

by the staff of Robot

An Epic Laser Battle BETWEEN ROBOTS

Behind the Scenes! How we created vision-based programming using FlowStone

Last month we introduced FlowStone, a new graphical programming language, in LERN. This issue we dig much deeper with a behind-the-scenes look into how the Lynxmotion BRAT biped and robot arm were programmed with FlowStone to make the featured video you can scan and watch now with your smartphone! We believe this new language from DSPRobotics showcases a new era in programming.

—Tom Atwood, editor-in-chief

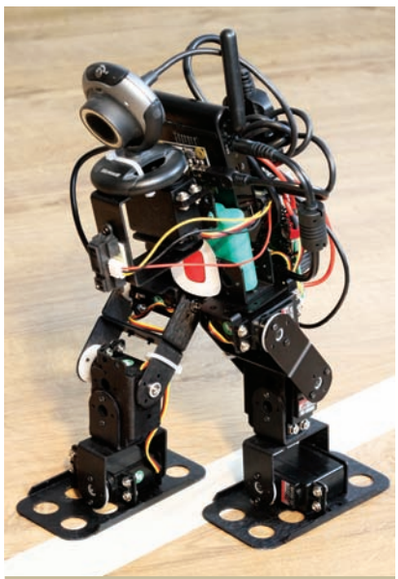
SEE THE VIDEO!

Scan this code on your smartphone with a bar reader app or type in find.botmag.com/051106

THE CHALLENGE:

A. Build a Biped Robot (The Intruder) that could follow a line on the ground using a simple webcam for its vision system.

B. Then build a Robot Arm (The Guard) that would protect a given area using another webcam for motion detection and targeting to aim a hi-powered laser (Pen) to defeat the biped intruder.



THE GEAR

FlowStone Professional V1.1.1

1 x LynxMotion Biped BRAT (The Intruder)

1 x LynxMotion Robot Arm (The Guard)

2 x Webcam (Vision system)

1 x Laser Pen (The Weapon)

1 x Sharp IR Sensor (The Target)

1 x Phidget Spatial 3/3/3 (For balancing the webcam platform)

1 x Phidget Relay Board (Laser Firing)

1 x Wireless USB Hub (BRAT - Coms)

2 x Laptops (The Brains - Intruder and Guard)

2 x USB to RS232 adapter

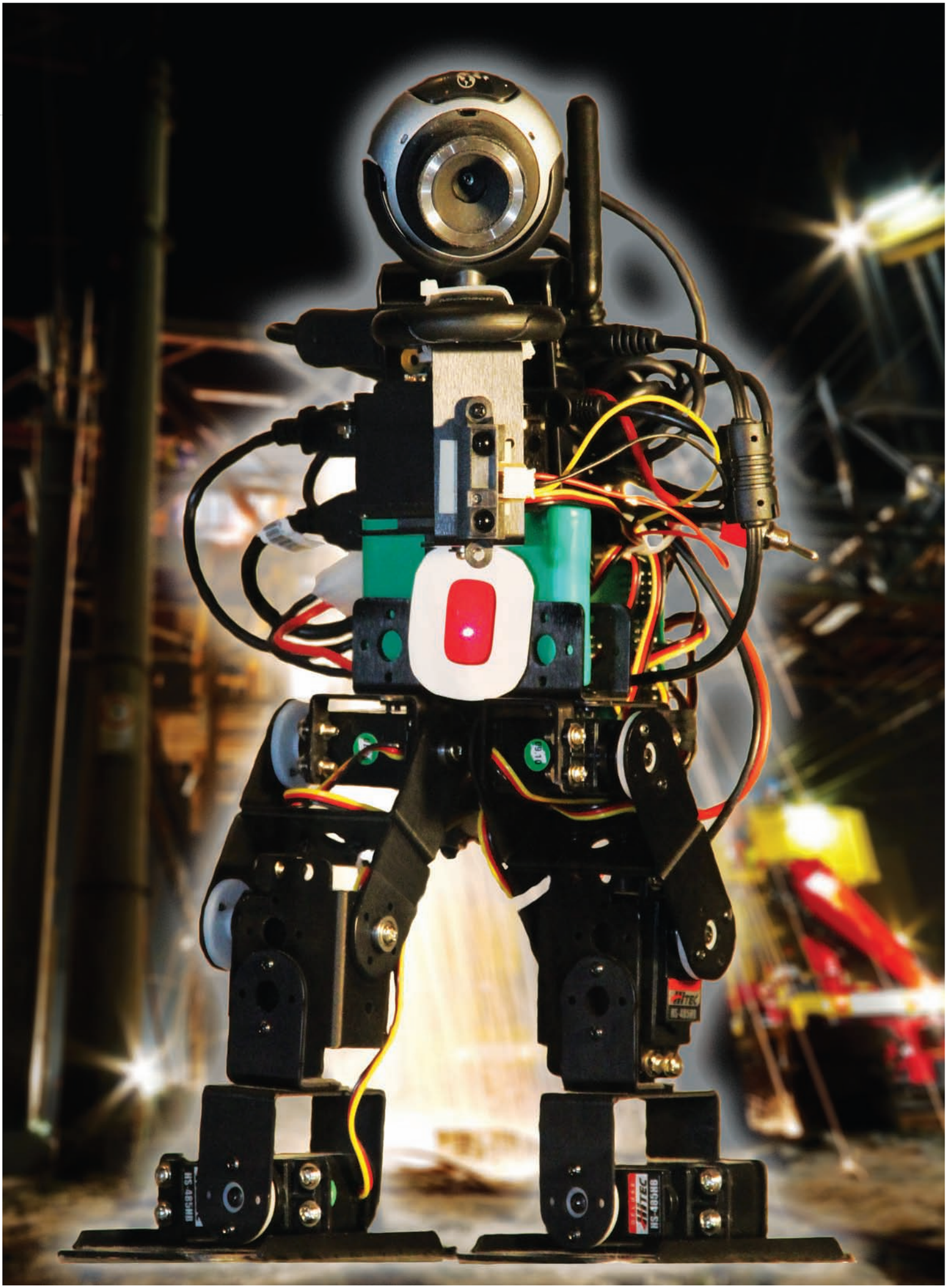
THE ROBOTS

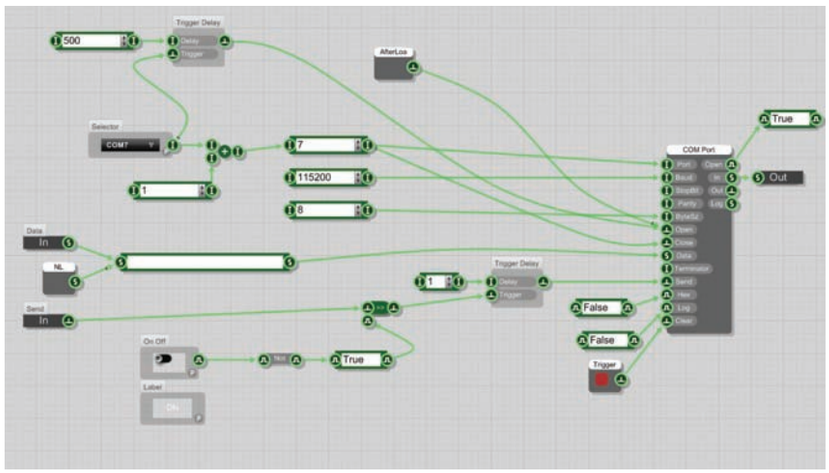
For our intruder we used a Biped BRAT from LynxMotion this is a great first step into robotics as it is relatively simple to program and it only has 6 degrees of freedom (DOF), or in layman's terms it has six servos to program. The interface is through a simple RS232 cable which we connected to a USB to RS232 convertor to connect to our laptop. In order to make our Brat wireless we also used a wireless USB Hub, this simply provides four USB ports remotely as if they were directly connected to the Laptop.

For our guard robot we used a LynxMotion Robot Arm, similar to the BRAT in as far as it too uses a SC-32 servo board, so our programming has some common elements. The main difference with the arm is that it has a fixed base and our challenge was to move the hand of the robot to a specific point using Inverse Kinematics (IK) to fire the laser.

SERVO CONTROL AND INTERFACING

The servo control was done using a PC COM port connected to the LynxMotion SC-32 servo board. In FlowStone you gain access to this by using the COM port module, in this case we used a baud rate of 115200, 8 data bits and Com7, which can be seen below:





The Com port module is set up to talk to the SC-32

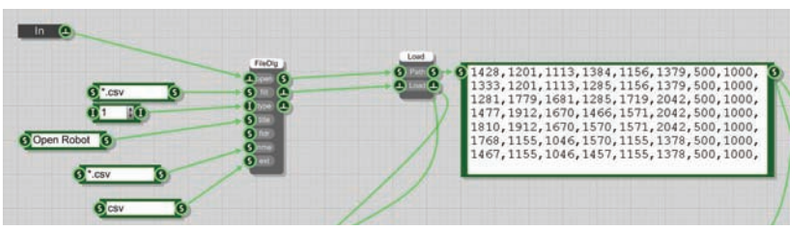
The serial data to control the servos is in the following format:
#0 P1482 #1 P1514 #2 P1412 #16 P1449 #17 P1589 #18 P1783 T1000 <CR>

Where:
# = Servo Number
P = Servo Position
T = Time to reach position
<CR> = Carriage Return or ASCI 13

BIPED MOVEMENT

The Biped Movements were based on the biped examples available on the DSPRobotics site, the 'BRAT Trainer', and the 'BRAT Sequencer'. These allow you to create sequences of movement for walking, turning left or right for example and then save each individual sequence to disk as a text file.

These were then loaded into the sequencer to produce the desired range of movements a bit like a macro: Walk 1 pace, Left a bit, Right a bit etc. In FlowStone you have full access to the PC file system so you can easily save and load files. The hardest part was creating the movement sequences in the first place, a little trial an error was needed to make the biped walk!



Here is an array of servo values loaded from a text file.

BIPED CAMERA STABILIZATION

When the biped walks it rocks side to side, which meant that sometimes the camera would loose sight of the line on the ground.

Phidget Spatial 3/3/3.

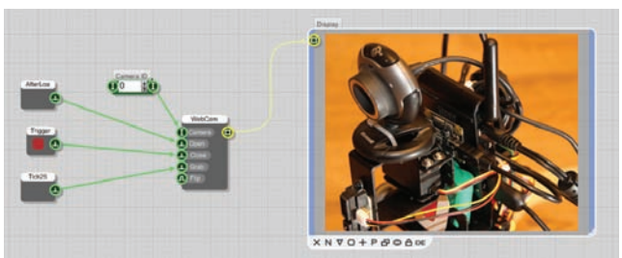
We compensated for this by mounting the webcam on a movable bracket and then used a Phidget Spatial 3/3/3 gyro accelerometer board to detect the movement of the robot. You can see this on the video as the robot rocks right the camera points left a little to compensate. However in the end we just captured the image when the robot was upright as this was the best image to calculate the line detection.

ARM MOVEMENT

Quite different from the biped the arm was far more sophisticated employing Inverse Kinematics (IK) to move the hand of the arm to a specific point in 3D space. Using the FlowStone IK module we were able to specify the lengths of each limb in the arm and the IK module did the rest. It calculated the correct angles to position the arm while keeping the hand level-since the gripper was gripping the webcam and we wanted to keep it level. So to move the arm we just needed to specify a single point and the arm would move to that position. So once the motion was detected we could wake up the arm and start scanning for targets.

VISION SYSTEM

The core of both applications was the use of standard webcams, one on the biped and one on the arm. The first task in both programs was to capture an image inside FlowStone, to do this we used the 'webcam' module. This is a standard module that enables access to any installed Windows Webcam device on your PC. The key to this module is the 'Grab' input which grabs an image from the webcam. We used a 'Tick25' module that generates a trigger signal 25 times a second and connected this to the grab input. The output is a bitmap which can be connected to a display module to test the operation of the webcams:



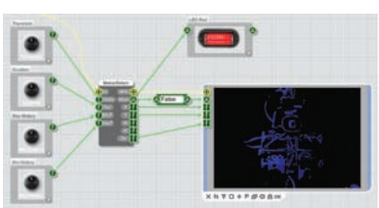
The FlowStone Webcam Module connected to a Display Module

COLOR DETECTION

The color detection was required on both the red target tracking and also the line following for the biped. The way this works it that you specify a target colour range using HSV (hue, saturation, and value) values, fortunately there was also a color to HSV module, as these values can be hard to calculate. There is also a minimum target area input, so it only picks targets of a certain size. There is a visual output showing the live video image plus any detected color targets highlighted in an on screen box. For our program we used the X,Y co-ordinates out to make decisions on what to do next.

MOTION DETECTION

Motion detection was used to wake up the arm and initiate the target tracking. Setting the Threshold, Duration



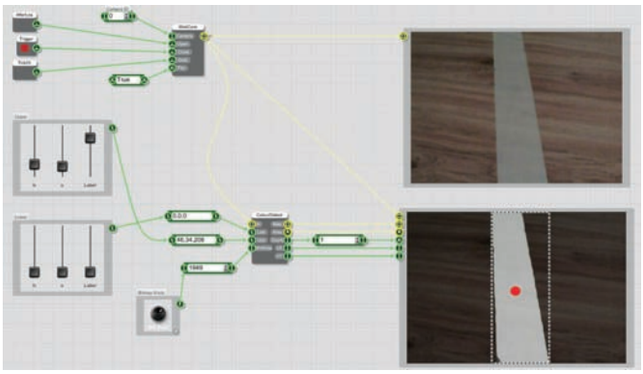
Here you can see the motion detected, highlighted in blue

and History you can fine tune how sensitive the detection is. You can even detect the motion direction if necessary so that you detect only objects moving in a certain directions! Once adequate motion is detected the motion output of the module goes from False to True. We used this output to turn on the red 'Motion Detected' LED.

BIPED LINE FOLLOWING

The line following for the biped was achieved using the color detect module to identify the path ahead and three simple biped movement sequences. Forward, Left a bit and Right a bit. Depending on the position of the line at the center balance point of the robot it would move accordingly.

The clever thing was that even though the line was at an angle the color detection calculated the center point of the actual line and not the center of the box defining the shape. This can be seen in the image below, look at where the red dot is compared to the bounding box:



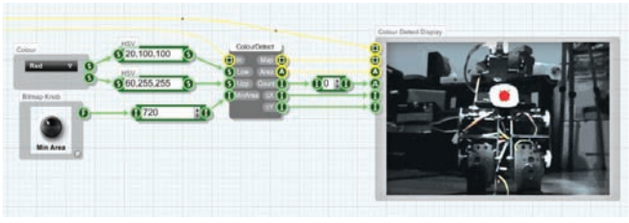
See the tape on the floor and the color detection highlighting the path ahead.

TARGET TRACKING

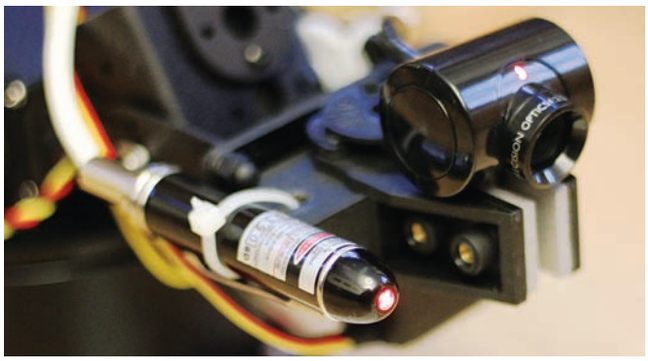
The colour detection identified where the bipeds red target was in the robot arms' field of view. The next thing was to rotate the base of the arm to track the target so that it was in the centre so the laser could be fired. For this we used the color detect X position out (cX). This gives the center point of the detected color region. We employed a simple algorithm of if the color detection was left, move the arm left a little, or if it was right move the arm right a little. If it was central stop and initiate the laser firing. In reality this worked surprisingly well, however if you wanted to make a more complete solution you could use Flowstone's PID (Proportional ,Integral, Derivative) module. This is a proper control system so the arm would move directly to the correct position for firing without any hunting around.

FIRING THE LASER

Once the arm had locked onto the biped using the colour detect the laser was fired. This was done using a Phidget 0/0/4 USB Relay board that simply closed a relay connected to the laser pens battery and thus firing the laser. All of the Phidgets devices are pre-programmed into FlowStone so it's just a case of drag and drop and wiring them up. The trigger signal was sent through a pulse module to create a one second firing pulse. This signal was also sent to an LED on the application GUI to indicate the firing of the Laser.



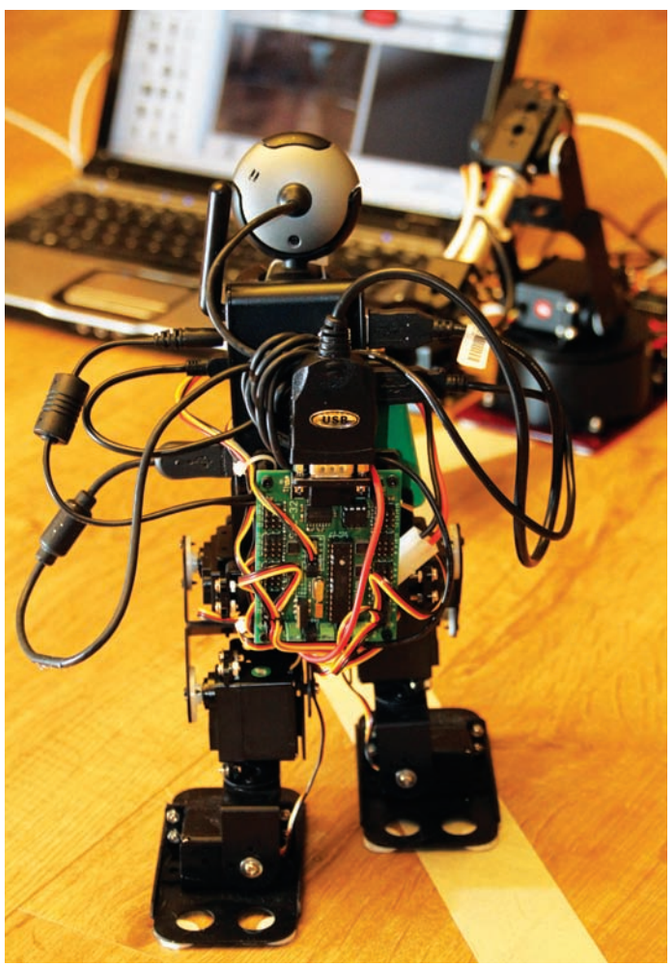
Here you can see the bipeds target 'lit up' ready for laser firing!



The Laser mounted on the arm together with the webcam.

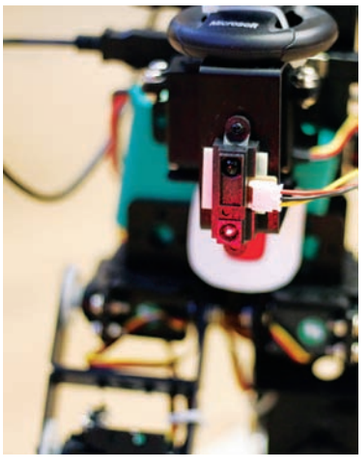
HOW TO KILL THE BIPED

Using the Sharp Infra Red sensor usually used for object detection, the analogue output from the sensor was connected to the analogue in on the LynxMotion SC-32 servo board. This voltage was then read inside FlowStone using the COM port and once over a set trigger level the command to kill all servos was given. This instantly removes the power from the servos so the robot falls to the ground. For a little more drama we delayed the kill signal a little so it was a slow death!



APPLICATION GUI

One of the powerful features of FlowStone is the ability to make you own custom GUI (Graphical User Interface). To do this there are lots of premade buttons and switches, LEDs etc. all of which are total editable and customizable, plus all of the usual drawing tools for lines and shapes etc. This means that you can import you own graphics and make a totally unique look for you final application. For this project we made our own background graphic with shaded windows for the video images, plus a custom LED, and switch

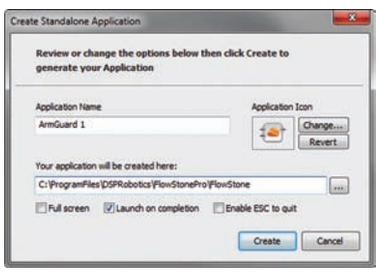


The laser hits the IR sensor.

Final custom GUI for the Arm Software.

MAKING AN EXE

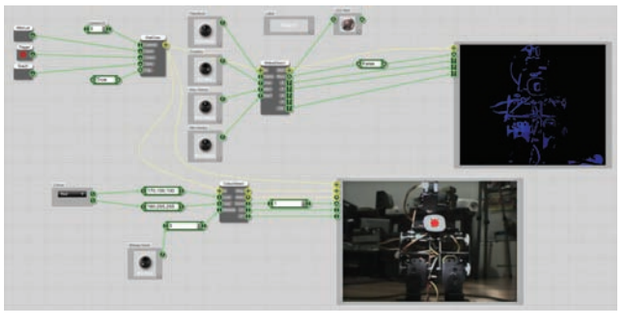
One unique feature of FlowStone is that once you are happy with you program running in the development environment. With a single click you can make you very own stand alone EXE program file. Some other graphical languages say that they make a stand-alone programs but they still need their runtime application installed in order for them to run a bit like Flash. This is not the case with FlowStone as it really does make a stand alone EXE! The great thing about this is that because of this feature you can choose to distribute or even sell you applications!



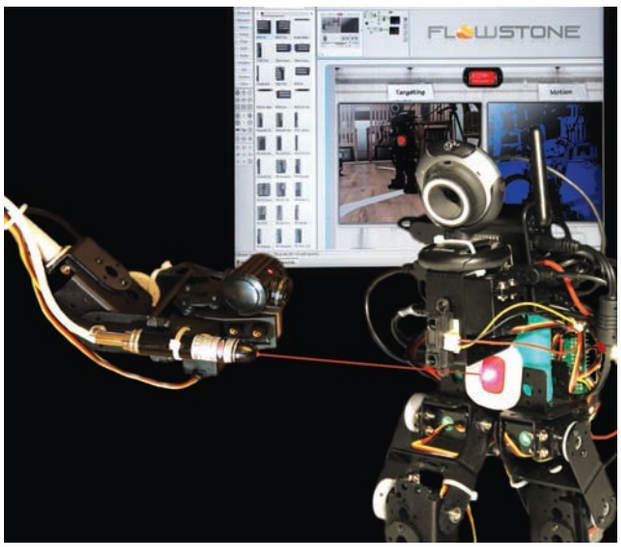
Simply make an EXE at the click of a mouse!

CONCLUSION

Putting everything together, once the biped was activated to attack in FlowStone, it would then follow the line of tape on the floor using the webcam as a guide, and also transmit the live video back to the base station (laptop) so



Tracking the target!



that we had front row seats. Once the Robot arm had detected the biped movement it would wake up and start tracking for a red target. Locked and loaded it would then fire the laser repeatedly until the target was hit. Once IR sensor had detected a hit from the laser the biped would slowly die and fall to the ground. Secure!

Now we have seen behind the scenes there is no doubt that FlowStone is a hugely powerful program allowing 1000's of application to be programmed quickly and efficiently. But make no mistake; even though FlowStone is a graphical programming environment, you still need to have some sort of programming background in order to get the most out of it. The main benefit is its scalability, and accessibility for all levels of programmer, you can start very simply and grow with the program. There is a FREE version of FlowStone available for Education which has all of the functions used in this project available to play with. The main limitation of FlowStone FREE is the project size and the inability to make EXE files. However there is a simple PAYG (pay as you go) system for low use users to make individual EXE file using the free version and chargeable Enterprise and Professional version to meet all levels. The possibilities of using FlowStone are endless and this is a complete and mature piece of software. We will certainly be using it again!

FlowStone FREE : FOC
FlowStone Enterprise: $296.00 US
FlowStone Professional: $745.00 US

Links
DSPRobotics, www.dsprobotics.com
Lynxmotion, www.lynxmotion.com, (866) 512-1024
Phidgets USA, www.phidgetsusa.com, (877) 898-1005

For more information, please see our source guide on page 89.