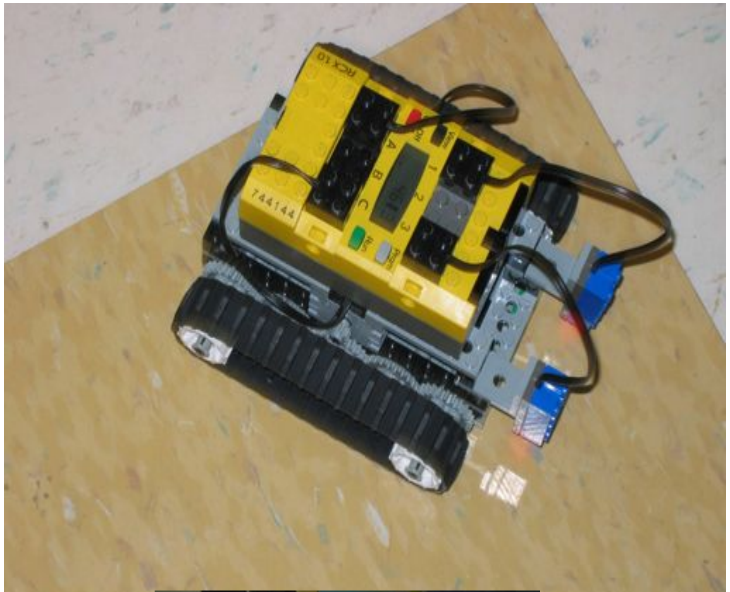
Robotics in the Classroom

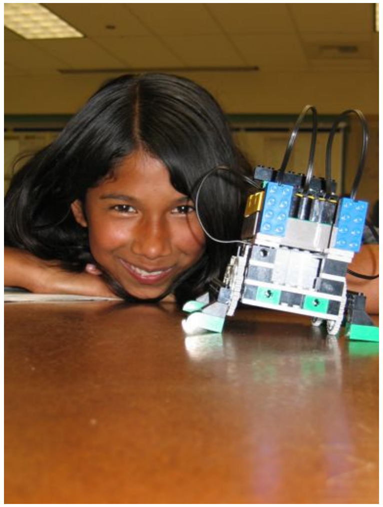
Our Experience with Robotics in the Classroom Using OpenSource Software

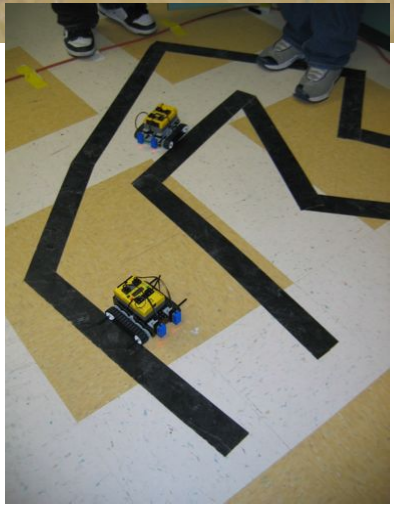
Eugene Clement

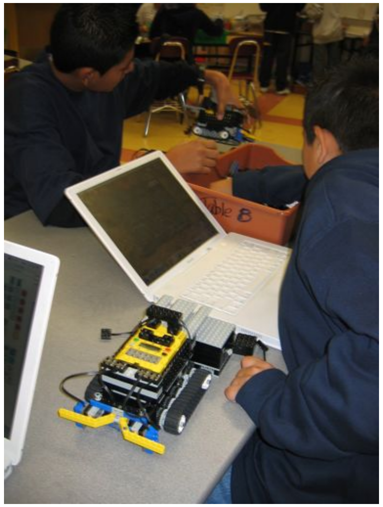
John Wise

Prequel: Before OpenSource
Lego Mindstorms and RoboLab on Macs





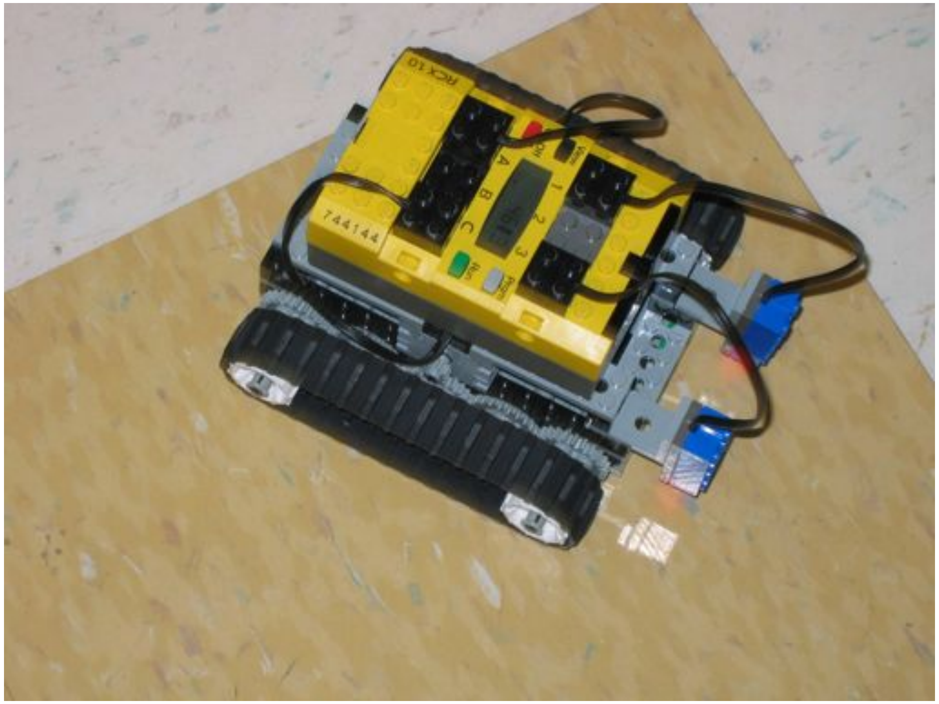




ROBOLAB
PROGRAMMING SOFTWARE
FOR THE LEGO RCX
LEGO dacta
Tufts University
College of Engineering
National Instruments
LabVIEW
LEGO MINDSTORMS
ROBOTICS
INVENTION SYSTEM
9719

Issues with Mindstorms/NXT

- Not Linux friendly
- Icon-based language
- Script-based languages (gcc and NQC) lack IDE, not beginner friendly
- Very limited access to internals
- 2 ports
- Limited upgrade path
- NXT less (old) Lego-like
- Expensive



Why OpenSource?

- Philosophy of School
- Cost
- Freedom
- Software Community
- School Demographics

Why Parallax BoeBot?

Welcome to the Parallax, Inc. Product CD-ROM

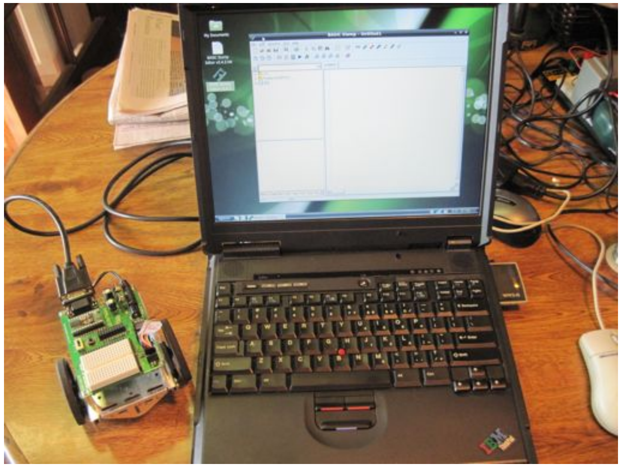
Please select an item to explore.

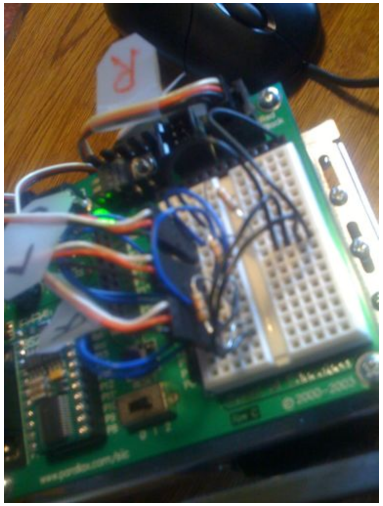
- Cost
- Educational Support
- Parallax IDE runs on WINE
- Internals accessible: pins available & breadboard
- Script-based programming

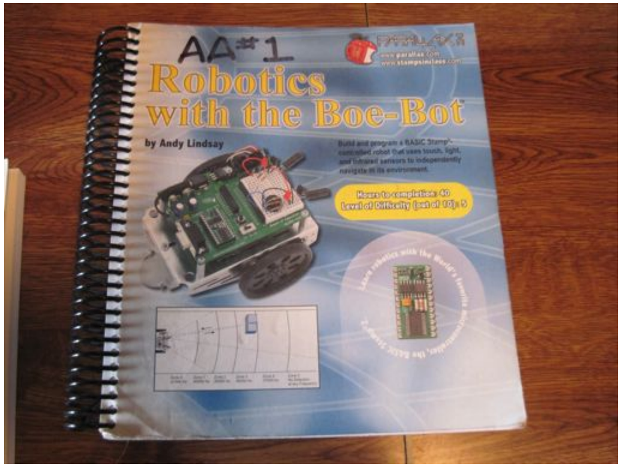
Why Thinkpads?

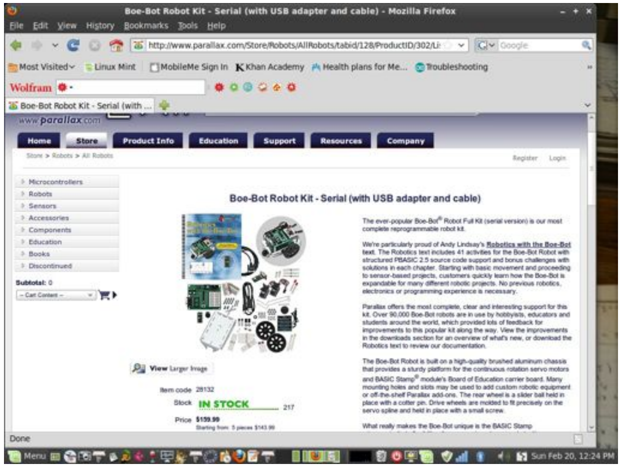
- Cost: Older a20m $50-$100 on eBay
- Availability: used company Thinkpads showing up on eBay and elsewhere continuously as companies upgrade
- Robust: drop them and they just might not break
- Variety: “newer” models start dropping into the affordable range
- Multiple Source: available everywhere
- Parts available: upgrade ram to 512mB, wireless cards
- Reliable
- No problem installing Linux
- Great keyboard!
- Dedicated to programming: don't share, limit wireless access to internet

BoeBot and pBasic









Boe-Bot Robot Kit - Serial (with USB adapter and cable) - Mozilla Firefox

http://www.parallax.com/Store/Robots/AllRobots/tabid/128/ProductID/302/L

Boe-Bot Robot Kit - Serial (with USB adapter and cable)

The ever-popular Boe-Bot® Robot Full Kit (serial version) is our most complete reprogrammable robot kit.

We're particularly proud of Andy Lindsay's **Robotics with the Boe-Bot** text. The Robotics text includes 41 activities for the Boe-Bot Robot with structured PBASIC 2.5 source code support and bonus challenges with solutions in each chapter. Starting with basic movement and proceeding to sensor-based projects, customers quickly learn how the Boe-Bot is expandable for many different robotic projects. No previous robotics, electronics or programming experience is necessary.

Parallax offers the most complete, clear and interesting support for this kit. Over 90,000 Boe-Bot robots are in use by hobbyists, educators and students around the world, which provided lots of feedback for improvements to this popular kit along the way. View the improvements in the downloads section for an overview of what's new, or download the Robotics text to review our documentation.

The Boe-Bot Robot is built on a high-quality brushed aluminum chassis that provides a sturdy platform for the continuous rotation servo motors and BASIC Stamp® module's Board of Education carrier board. Many mounting holes and slots may be used to add custom robotic equipment or off-the-shelf Parallax add-ons. The rear wheel is a slider ball held in place with a cotter pin. Drive wheels are molded to fit precisely on the servo spline and held in place with a small screw.

What really makes the Boe-Bot unique is the BASIC Stamp

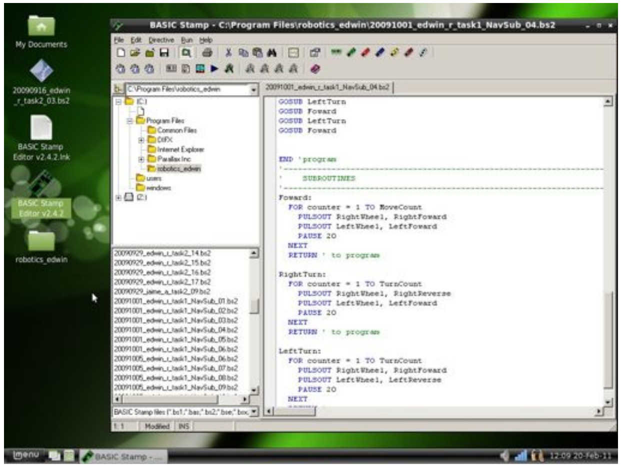
Subtotal: 0

Item code 28132

Stock IN STOCK 217

Price $159.99

Starting from: 5 pieces $143.99



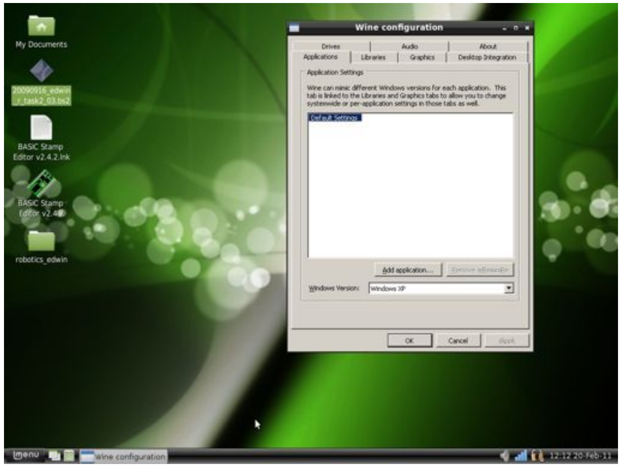
BASIC Stamp - C:\Program Files\robotics_edwin\20091001_edwin_r_task1_NavSub_04.bs2

My Documents
20090916_edwin_r_task2_03.bs2
BASIC Stamp Editor v2.4.2.lnk
BASIC Stamp Editor v2.4.2
robotics_edwin
File Edit Directive Run Help
C:\Program Files\robotics_edwin
(C:)
Program Files
Common Files
DIFX
Internet Explorer
Parallax Inc
robotics_edwin
users
windows
(Z:)
BASIC Stamp files (*.bs1,*.bas,*.bs2,*.bse,*.bsx
1: 1
Modified
INS
menu
BASIC Stamp - ...
12:09 20-Feb-11

20090929_edwin_r_task2_14.bs2
20090929_edwin_r_task2_15.bs2
20090929_edwin_r_task2_16.bs2
20090929_edwin_r_task2_17.bs2
20090929_jaime_a_task2_09.bs2
20091001_edwin_r_task1_NavSub_01.bs2
20091001_edwin_r_task1_NavSub_02.bs2
20091001_edwin_r_task1_NavSub_03.bs2
20091001_edwin_r_task1_NavSub_04.bs2
20091001_edwin_r_task1_NavSub_05.bs2
20091001_edwin_r_task1_NavSub_06.bs2
20091005_edwin_r_task1_NavSub_06.bs2
20091005_edwin_r_task1_NavSub_07.bs2
20091005_edwin_r_task1_NavSub_08.bs2
20091005_edwin_r_task1_NavSub_09.bs2

20091001_edwin_r_task1_NavSub_04.bs2

```
GOSUB LeftTurn
GOSUB Foward
GOSUB LeftTurn
GOSUB Foward


END 'program
'-----------------------------------------------------------------
'    SUBROUTINES
'-----------------------------------------------------------------
Foward:
  FOR counter = 1 TO MoveCount
    PULSOUT RightWheel, RightFoward
    PULSOUT LeftWheel, LeftFoward
    PAUSE 20
  NEXT
  RETURN ' to program

RightTurn:
  FOR counter = 1 TO TurnCount
    PULSOUT RightWheel, RightReverse
    PULSOUT LeftWheel, LeftFoward
    PAUSE 20
  NEXT
  RETURN ' to program

LeftTurn:
  FOR counter = 1 TO TurnCount
    PULSOUT RightWheel, RightFoward
    PULSOUT LeftWheel, LeftReverse
    PAUSE 20
  NEXT
```



Wine configuration

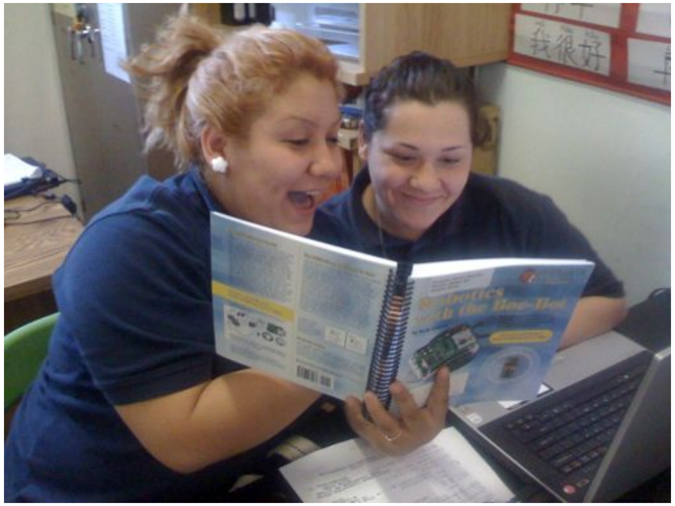
Application Settings

Wine can mimic different Windows versions for each application. This tab is linked to the Libraries and Graphics tabs to allow you to change systemwide or per-application settings in those tabs as well.

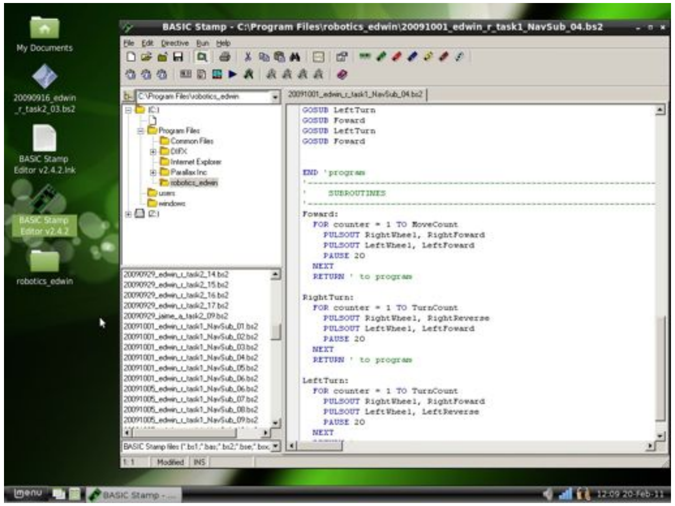
Default Settings

Windows Version: Windows XP

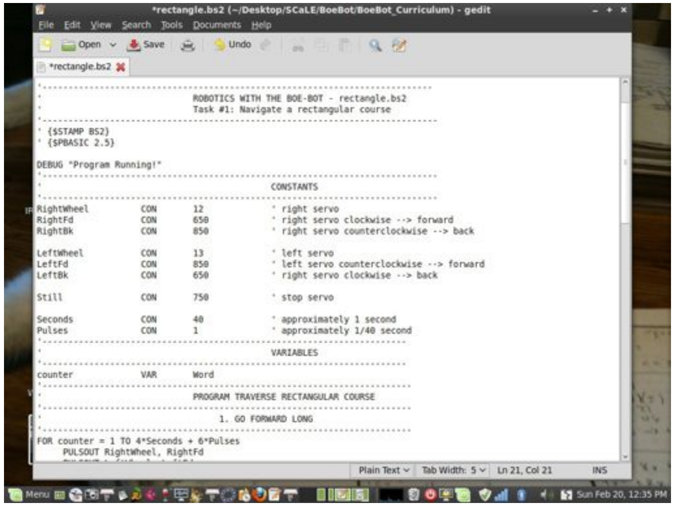
Programming



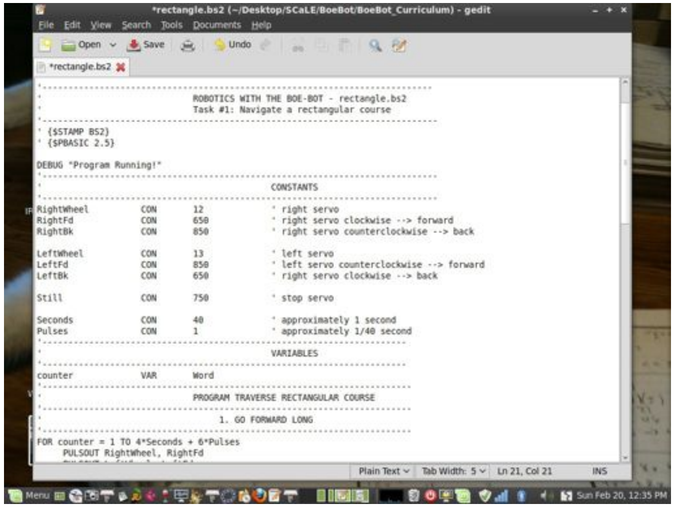
pBasic Programs


BASIC Stamp - C:\Program Files\robotics_edwin\20091001_edwin_r_task1_NavSub_04.bs2
File Edit Directive Run Help
20091001_edwin_r_task1_NavSub_04.bs2
My Documents
BASIC Stamp Editor v2.4.2
robotics_edwin
12:09 20-Feb-11

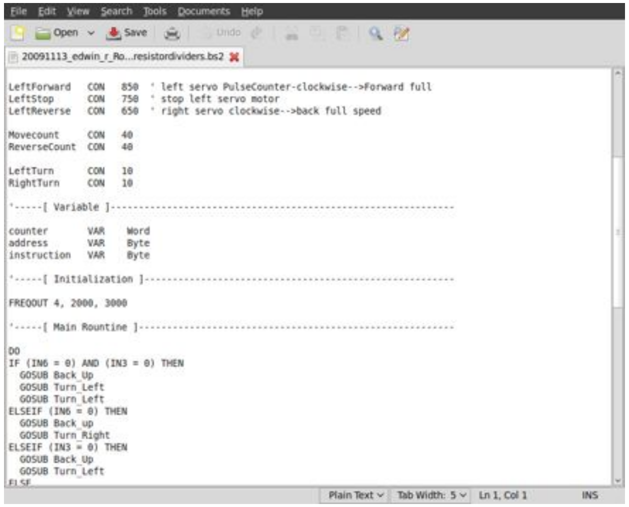
```
GOSUB LeftTurn
GOSUB Foward
GOSUB LeftTurn
GOSUB Foward


END 'program
'-----------------------------------------------------------------------
'     SUBROUTINES
'-----------------------------------------------------------------------
Foward:
  FOR counter = 1 TO MoveCount
    PULSOUT RightWheel, RightFoward
    PULSOUT LeftWheel, LeftFoward
    PAUSE 20
  NEXT
  RETURN ' to program

RightTurn:
  FOR counter = 1 TO TurnCount
    PULSOUT RightWheel, RightReverse
    PULSOUT LeftWheel, LeftFoward
    PAUSE 20
  NEXT
  RETURN ' to program

LeftTurn:
  FOR counter = 1 TO TurnCount
    PULSOUT RightWheel, RightFoward
    PULSOUT LeftWheel, LeftReverse
    PAUSE 20
  NEXT
```


*rectangle.bs2 (~/Desktop/SCaLE/BoeBot/BoeBot_Curriculum) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
*rectangle.bs2
Plain Text
Tab Width: 5
Ln 21, Col 21
INS
Menu
Sun Feb 20, 12:35 PM

```
'---------------------------------------------------------------------------
'                            ROBOTICS WITH THE BOE-BOT - rectangle.bs2
'                            Task #1: Navigate a rectangular course
'---------------------------------------------------------------------------
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Program Running!"
'---------------------------------------------------------------------------
'                                          CONSTANTS
'---------------------------------------------------------------------------
RightWheel          CON       12             ' right servo
RightFd             CON       650            ' right servo clockwise --> forward
RightBk             CON       850            ' right servo counterclockwise --> back

LeftWheel           CON       13             ' left servo
LeftFd              CON       850            ' left servo counterclockwise --> forward
LeftBk              CON       650            ' right servo clockwise --> back

Still               CON       750            ' stop servo

Seconds             CON       40             ' approximately 1 second
Pulses              CON       1              ' approximately 1/40 second
'-------------------------------------------------------------------------
'                                          VARIABLES
'-------------------------------------------------------------------------
counter             VAR       Word
'-------------------------------------------------------------------------
'                            PROGRAM TRAVERSE RECTANGULAR COURSE
'-------------------------------------------------------------------------
'                                 1. GO FORWARD LONG
'-------------------------------------------------------------------------
FOR counter = 1 TO 4*Seconds + 6*Pulses
     PULSOUT RightWheel, RightFd
```

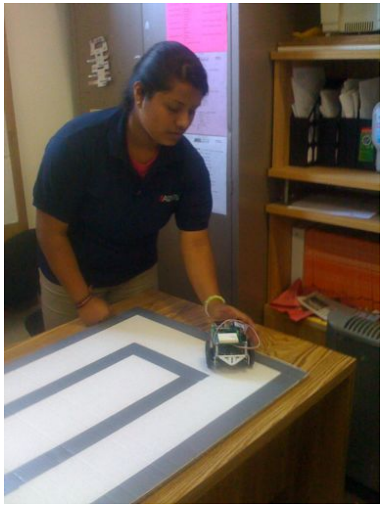

*rectangle.bs2 (~/Desktop/SCaLE/BoeBot/BoeBot_Curriculum) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
*rectangle.bs2
Plain Text
Tab Width: 5
Ln 21, Col 21
INS
Menu
Sun Feb 20, 12:35 PM

```
'---------------------------------------------------------------------------
'                            ROBOTICS WITH THE BOE-BOT - rectangle.bs2
'                            Task #1: Navigate a rectangular course
'---------------------------------------------------------------------------
' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "Program Running!"
'---------------------------------------------------------------------------
'                                          CONSTANTS
'---------------------------------------------------------------------------
RightWheel          CON       12             ' right servo
RightFd             CON       650            ' right servo clockwise --> forward
RightBk             CON       850            ' right servo counterclockwise --> back

LeftWheel           CON       13             ' left servo
LeftFd              CON       850            ' left servo counterclockwise --> forward
LeftBk              CON       650            ' right servo clockwise --> back

Still               CON       750            ' stop servo

Seconds             CON       40             ' approximately 1 second
Pulses              CON       1              ' approximately 1/40 second
'-------------------------------------------------------------------------
'                                          VARIABLES
'-------------------------------------------------------------------------
counter             VAR       Word
'-------------------------------------------------------------------------
'                            PROGRAM TRAVERSE RECTANGULAR COURSE
'-------------------------------------------------------------------------
'                                 1. GO FORWARD LONG
'-------------------------------------------------------------------------
FOR counter = 1 TO 4*Seconds + 6*Pulses
     PULSOUT RightWheel, RightFd
```

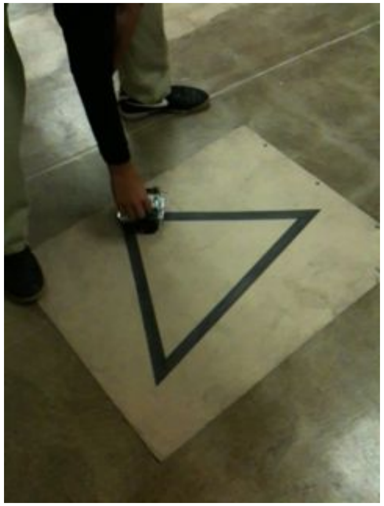

File Edit View Search Tools Documents Help
Open Save Undo
20091113_edwin_r_Ro...resistordividers.bs2
Plain Text
Tab Width: 5
Ln 1, Col 1
INS

```
LeftForward   CON   850  ' left servo PulseCounter-clockwise-->Forward full
LeftStop      CON   750  ' stop left servo motor
LeftReverse   CON   650  ' right servo clockwise-->back full speed

Movecount     CON   40
ReverseCount  CON   40

LeftTurn      CON   10
RightTurn     CON   10

'-----[ Variable ]----------------------------------------------------------

counter       VAR    Word
address       VAR    Byte
instruction   VAR    Byte

'-----[ Initialization ]----------------------------------------------------

FREQOUT 4, 2000, 3000

'-----[ Main Rountine ]-----------------------------------------------------

DO
IF (IN6 = 0) AND (IN3 = 0) THEN
  GOSUB Back_Up
  GOSUB Turn_Left
  GOSUB Turn_Left
ELSEIF (IN6 = 0) THEN
  GOSUB Back_up
  GOSUB Turn_Right
ELSEIF (IN3 = 0) THEN
  GOSUB Back_Up
  GOSUB Turn_Left
ELSE
```

Curriculum



Curriculum



Semester Exam: hard-coded navigation

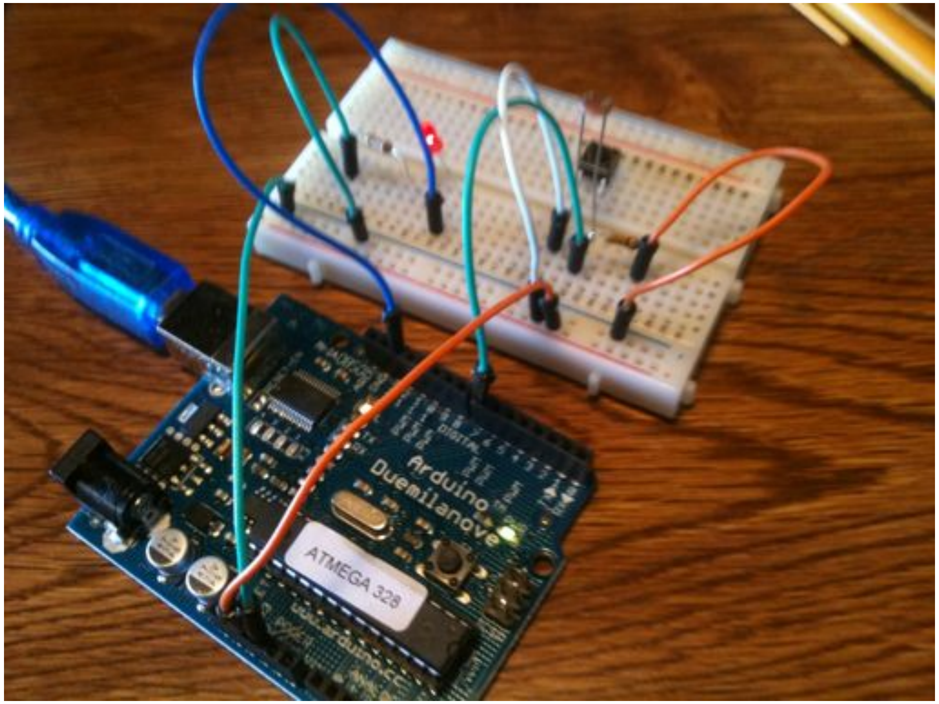
Final Exam: sensor-based navigation

Issues with BoeBots

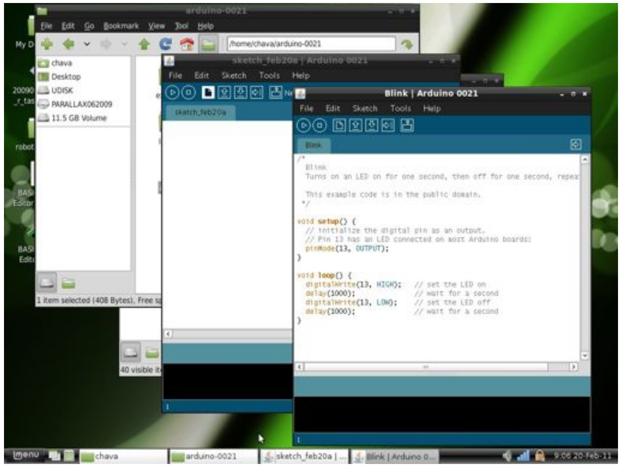
- pBasic lacked functions, couldn't pass parameters to subroutines
- Didn't run native on Linux
- Wine added extra level of complexity
- Text not well rendered
- Some problems with IDE under Wine
- Wanted students to have closer exposure to industry-standard language
- Wanted more open architecture
- Wanted more OpenSource community resources
- Wanted more upgrade paths
- Expense, always!

Why Arduino?

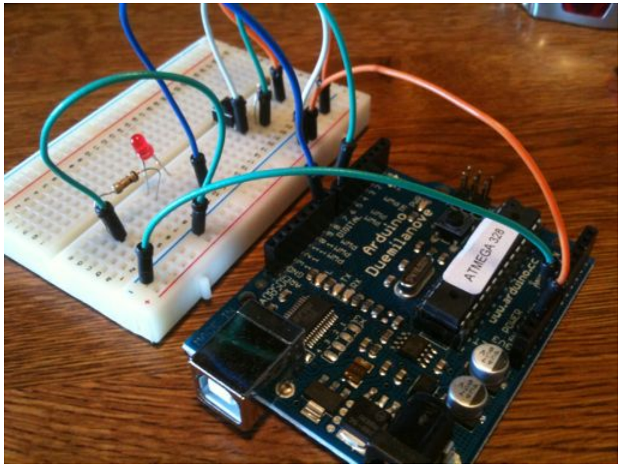
- Open Source Software
- Open Source Hardware
- Cost
- Beginner Friendly IDE
- Runs on Linux
- Online resources
- Friendly C++ programming
- Example code in IDE
- Build on BoeBot experience
- Lots of paths of exploration besides robotics

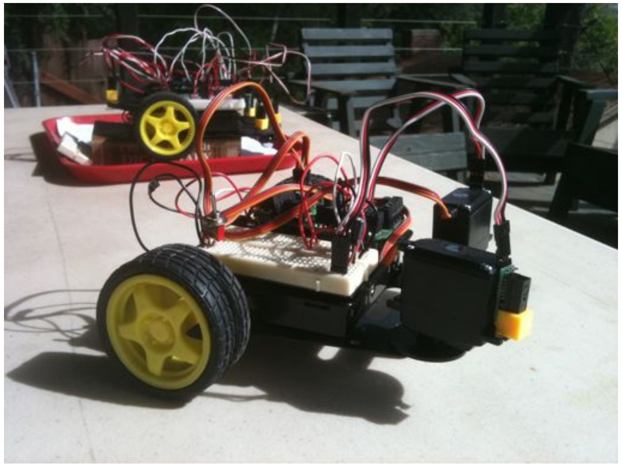


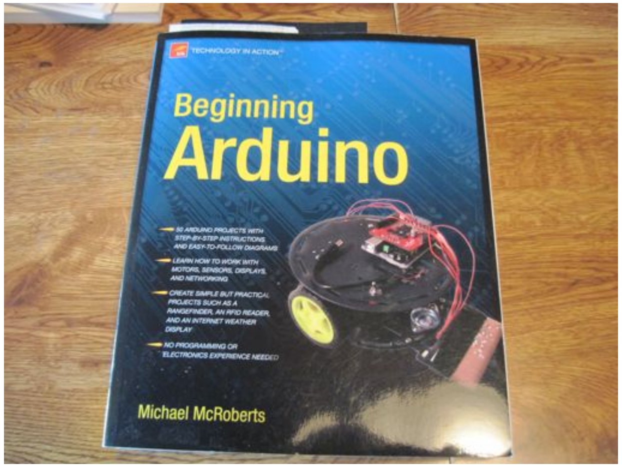
Arduino

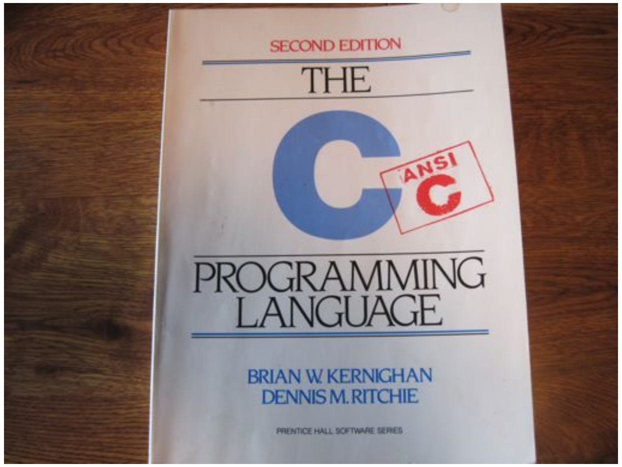


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

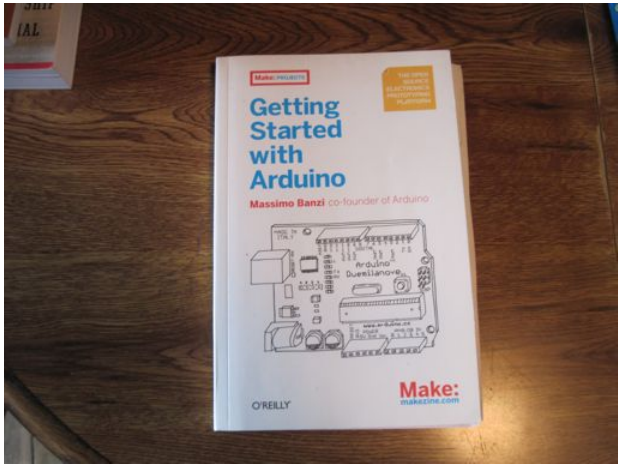


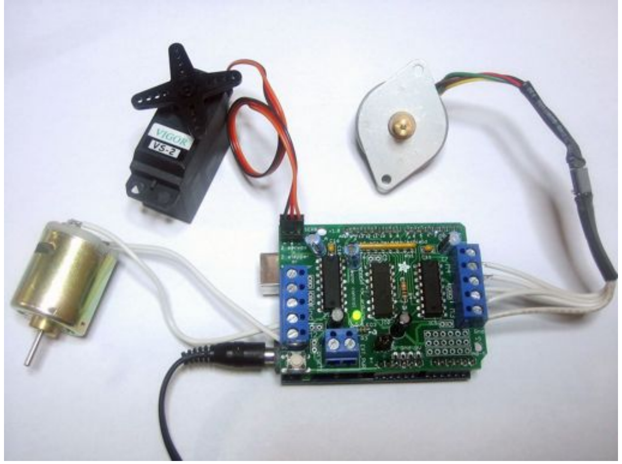


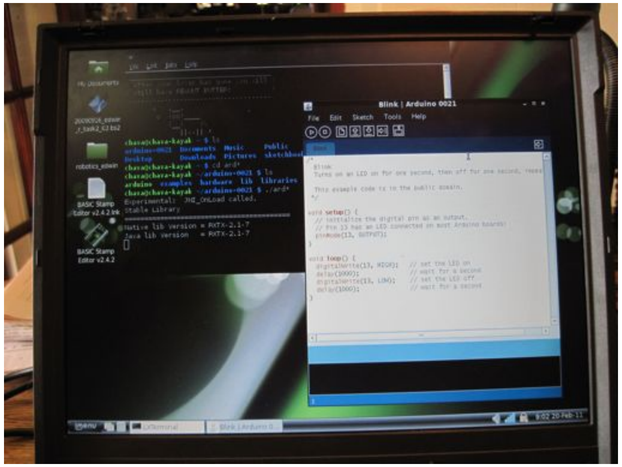

TECHNOLOGY IN ACTION
Beginning
Arduino
50 ARDUINO PROJECTS WITH STEP-BY-STEP INSTRUCTIONS AND EASY-TO-FOLLOW DIAGRAMS
LEARN HOW TO WORK WITH MOTORS, SENSORS, DISPLAYS, AND NETWORKING
CREATE SIMPLE BUT PRACTICAL PROJECTS SUCH AS A RANGEFINDER, AN RFID READER, AND AN INTERNET WEATHER DISPLAY
NO PROGRAMMING OR ELECTRONICS EXPERIENCE NEEDED
Michael McRoberts

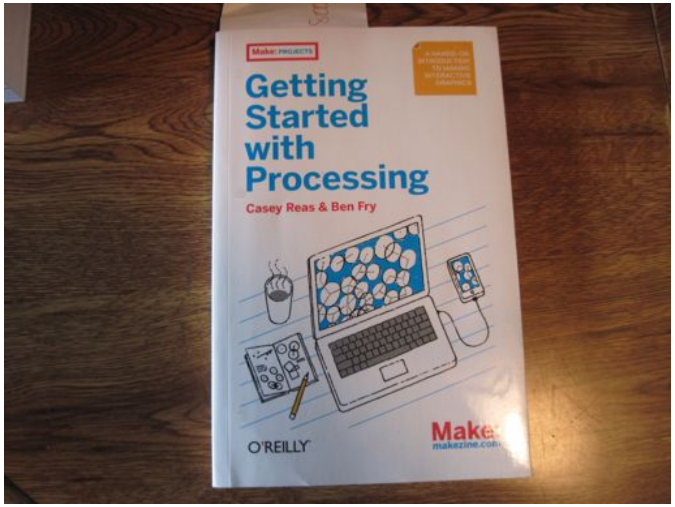


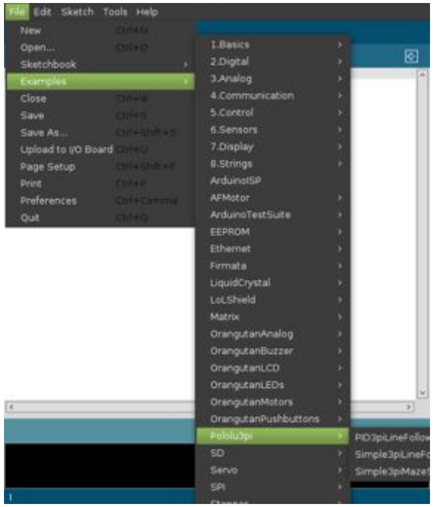
Arduino

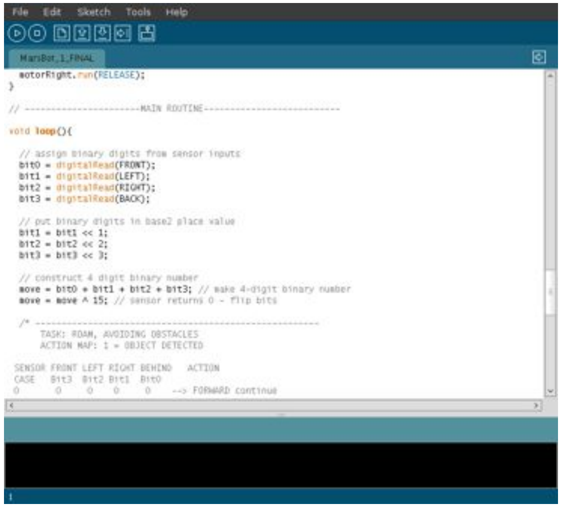

Make: PROJECTS
THE OPEN SOURCE ELECTRONICS PROTOTYPING PLATFORM
Getting Started with Arduino
Massimo Banzi co-founder of Arduino
O'REILLY
Make:
makezine.com






Make: PROJECTS
A HANDS-ON INTRODUCTION TO MAKING INTERACTIVE GRAPHICS
Getting Started with Processing
Casey Reas & Ben Fry
O'REILLY
Make:
makezine.com





```
  motorRight.run(RELEASE);
}

// ----------------------MAIN ROUTINE--------------------------

void loop(){

  // assign binary digits from sensor inputs
  bit0 = digitalRead(FRONT);
  bit1 = digitalRead(LEFT);
  bit2 = digitalRead(RIGHT);
  bit3 = digitalRead(BACK);

  // put binary digits in base2 place value
  bit1 = bit1 << 1;
  bit2 = bit2 << 2;
  bit3 = bit3 << 3;

  // construct 4 digit binary number
  move = bit0 + bit1 + bit2 + bit3; // make 4-digit binary number
  move = move ^ 15; // sensor returns 0 - flip bits

  /* ------------------------------------------------------
     TASK: ROAM, AVOIDING OBSTACLES
     ACTION MAP: 1 = OBJECT DETECTED

 SENSOR FRONT LEFT RIGHT BEHIND   ACTION
 CASE   Bit3  Bit2 Bit1  Bit0
 0       0     0    0     0    --> FORWARD continue
```

Cleaning, Climbing, Crawling Robots

SERVO

SERVO MAG

FOR THE ROBOT INNOVATOR

www.servomagazine.com

Building The ARDUINO BOT

You've met the brain, Now build the base!

+ P
Th
What can
chip do for yo

+ VEX Motor Control Experiments

Use simple VEX hardware to aim a solar panel and track the sun.

Making Robots With The Arduino

Part 2

By Gordon McCom

The ArdBot is a low-cost, 7" diameter servo-driven robot base, ready for expansion. It's called ArdBot because it's based on the popular and inexpensive Arduino microcontroller board. The ArdBot costs under $80 to build; even less if you already have some of the components, like the breadboard, jumper wires, and battery holder.

ArdBot Basic Design

About the Servo Drive

Making the ArdBot Base

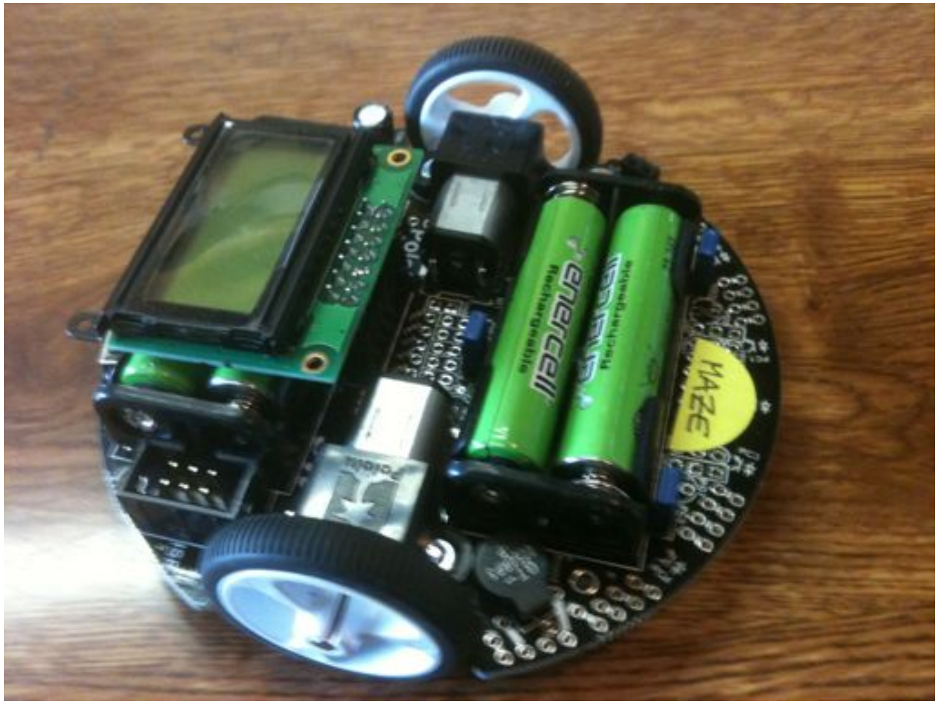
Table 1. Mechanical Parts.

Table 2. Motors and Wheels.

Issues with Arduino

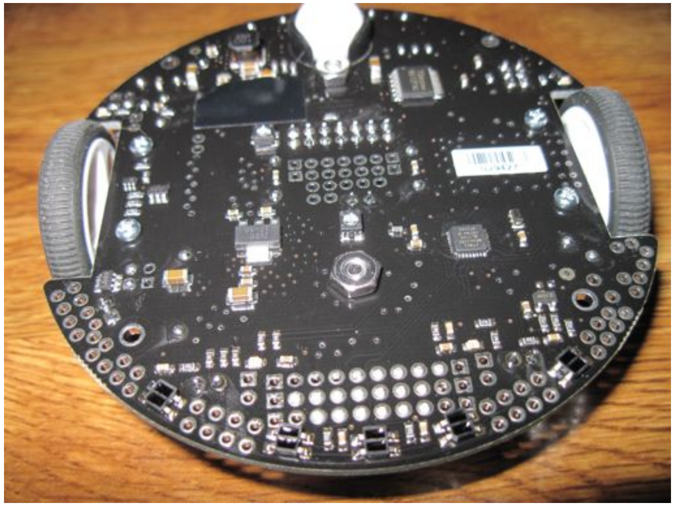
- Caveat: experience is not classroom tested
- Building a very beginner-friendly IDE has a cost: problem writing to serial monitor
- Hiding complexity introduces more complexity: Arduino introduces wrapper around C++; built on top of Processor, on top of Java to run on multiple platforms (including Linux)
- Beginner friendliness, OpenSource community, example code in IDE, open hardware paths, openness to standard C++ programming...all outweigh concerns

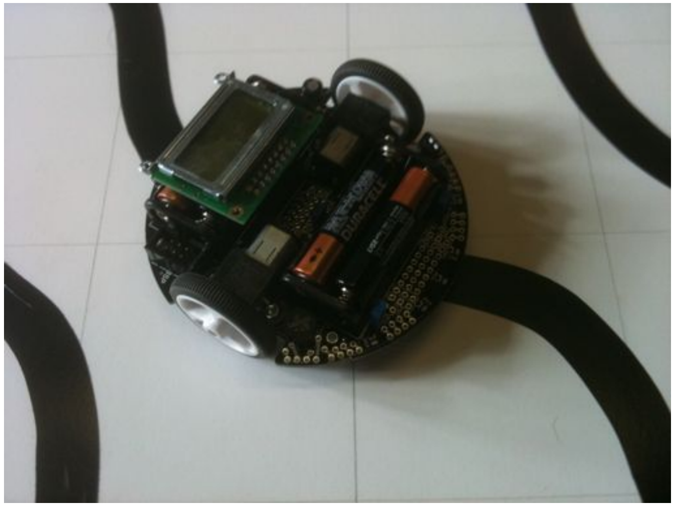
Pololu 3pi



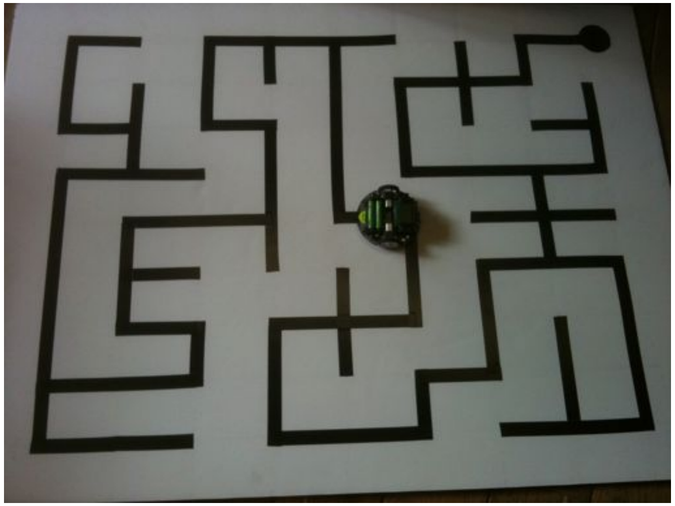
- Can be programmed with Arduino IDE
- Need an AVR ISP cable
- Pololu supplies library of Arduino wrappers
- Great sensors
- Next level of sophistication
- Serious C++ programming
- Line follower/Maze solver
- Decks can be added with additional sensors

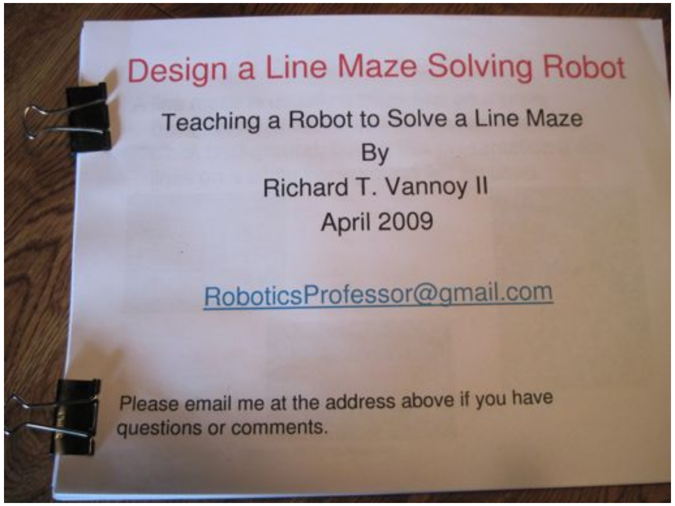
3pi Line Follower





Maze Solving



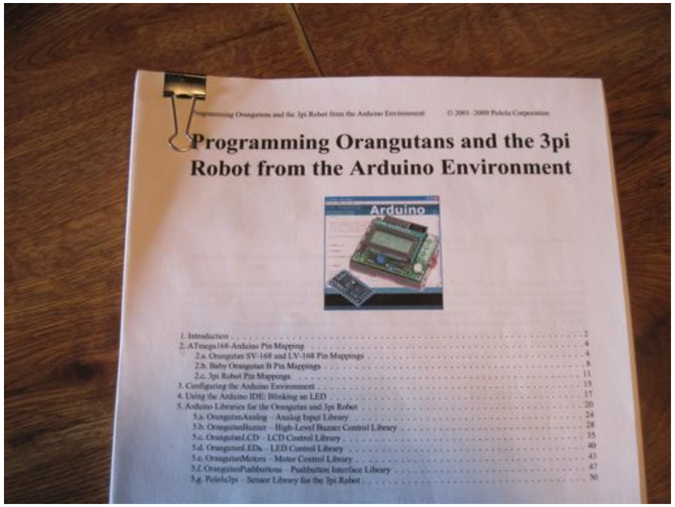


Design a Line Maze Solving Robot

Teaching a Robot to Solve a Line Maze
By
Richard T. Vannoy II
April 2009

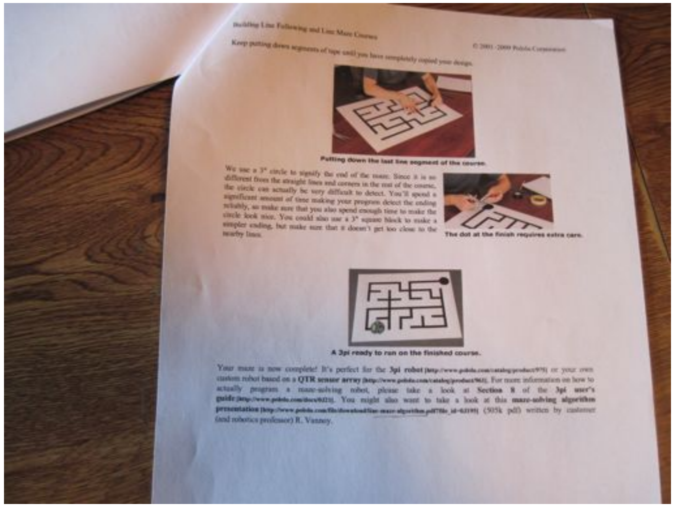
RoboticsProfessor@gmail.com

Please email me at the address above if you have questions or comments.



Programming Orangutans and the 3pi Robot from the Arduino Environment © 2001–2009 Pololu Corporation

Programming Orangutans and the 3pi Robot from the Arduino Environment



Building Line Following and Line Maze Courses © 2001–2009 Pololu Corporation

Keep putting down segments of tape until you have completely copied your design.

Putting down the last line segment of the course.

We use a 3" circle to signify the end of the maze. Since it is so different from the straight lines and corners in the rest of the course, the circle can actually be very difficult to detect. You'll spend a significant amount of time making your program detect the ending reliably, so make sure that you also spend enough time to make the circle look nice. You could also use a 3" square block to make a simpler ending, but make sure that it doesn't get too close to the nearby lines.

The dot at the finish requires extra care.

A 3pi ready to run on the finished course.

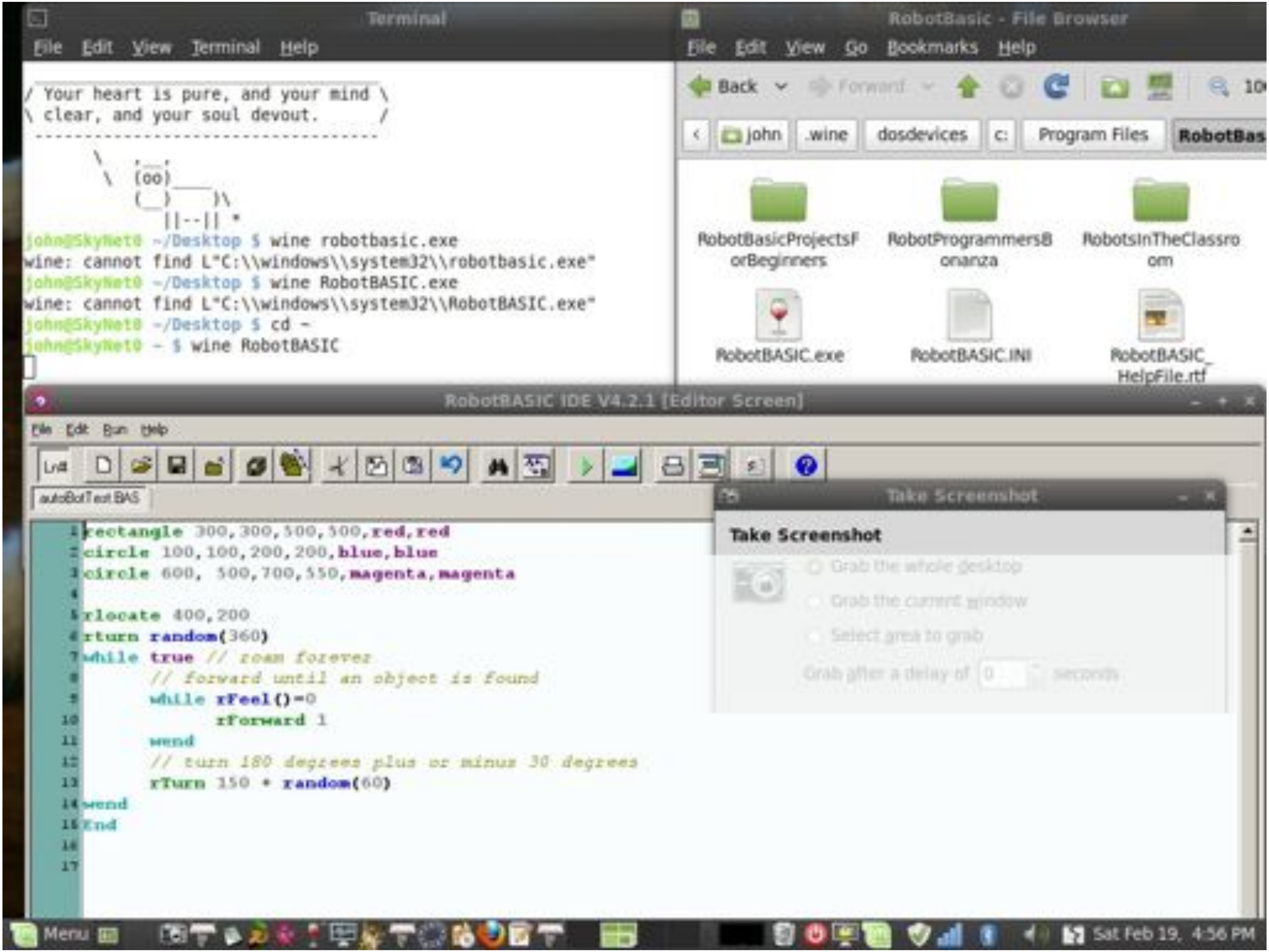
Your maze is now complete! It's perfect for the **3pi robot** [http://www.pololu.com/catalog/product/975] or your own custom robot based on a **QTR sensor array** [http://www.pololu.com/catalog/product/961]. For more information on how to actually program a maze-solving robot, please take a look at **Section 8** of the **3pi user's guide** [http://www.pololu.com/docs/0J21]. You might also want to take a look at this **maze-solving algorithm presentation** [http://www.pololu.com/file/download/line-maze-algorithm.pdf?file_id=0J195] (505k pdf) written by customer (and robotics professor) R. Vanney.

3pi Maze Solver

RobotBASIC?

RobotBASIC Projects For Beginners
John Blankenship
Samuel Mishal
Robots in the Classroom
Learn to Program with RobotBASIC
John Blankenship
Samuel Mishal
ROBOT PROGRAMMER'S BONANZA
JOHN BLANKENSHIP AND SAMUEL MISHAL
HARDWARE INTERFACING WITH ROBOTBASIC
THE FUNDAMENTALS
JOHN BLANKENSHIP AND SAMUEL MISHAL
Enhancing the Pololu 3pi with RobotBASIC
John Blankenship and Samuel Mishal

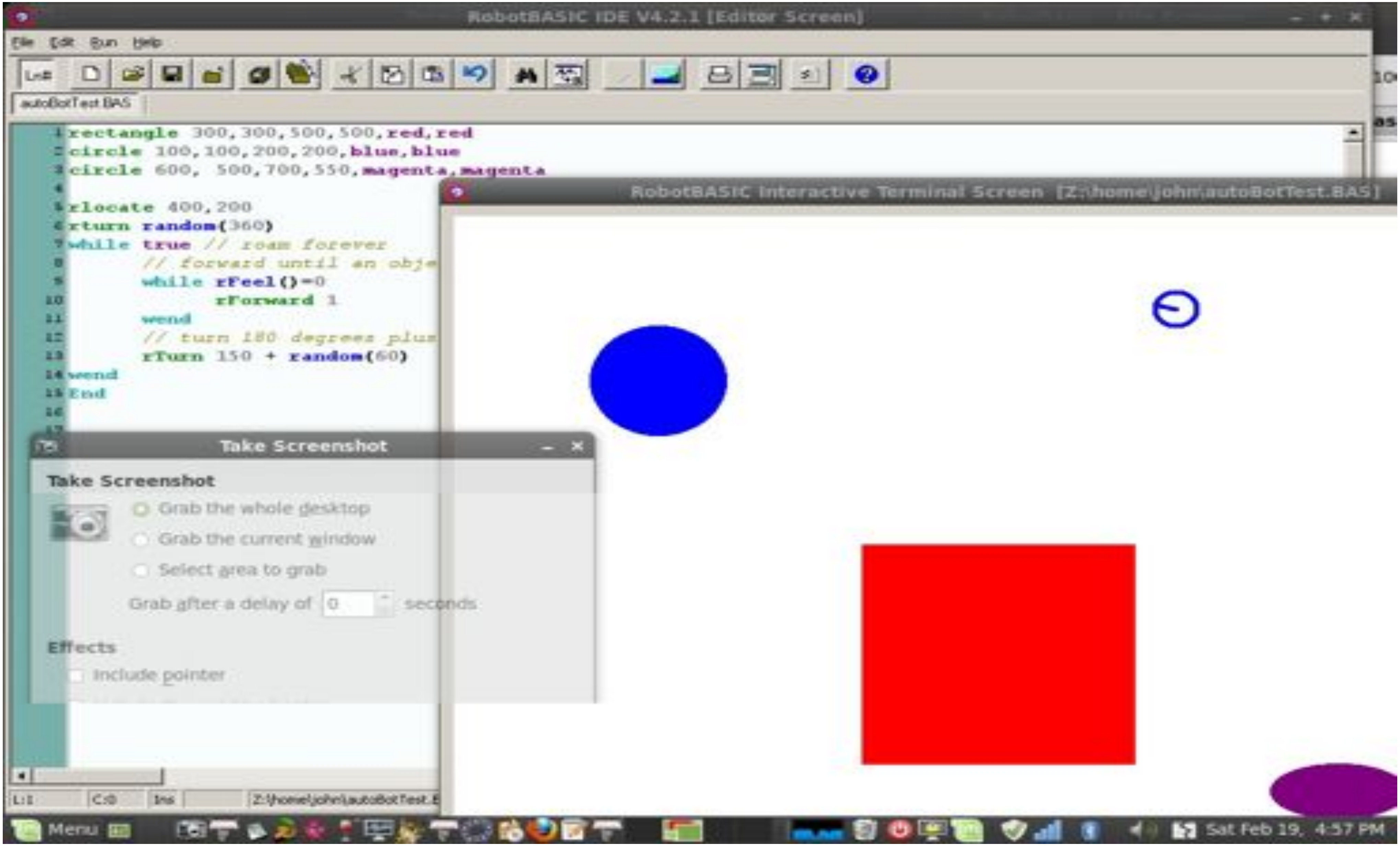
- Free software downloaded from website
- IDE runs under WINE
- Spans the gap from elementary school and no robots to advanced high school with BoeBots and Pololu 3pi's
- Simulator:
- Logo-like, virtual robot on screen
- Clean full-featured language
- Variety of sensors for “turtle” implemented
- Virtual machine implemented on Parallax BoeBot and Pololu 3pi
- Simulator code runs on targeted hardware (have to build sensor arrays on hardware platforms to conform with virtual robots)
- Next area of investigation



```
 ________________________________
/ Your heart is pure, and your mind \
\ clear, and your soul devout.      /
 ---------------------------------
        \   ,__,
         \  (oo)____
            (__)    )\
               ||--|| *
john@SkyNet0 ~/Desktop $ wine robotbasic.exe
wine: cannot find L"C:\\windows\\system32\\robotbasic.exe"
john@SkyNet0 ~/Desktop $ wine RobotBASIC.exe
wine: cannot find L"C:\\windows\\system32\\RobotBASIC.exe"
john@SkyNet0 ~/Desktop $ cd ~
john@SkyNet0 ~ $ wine RobotBASIC
```

```
rectangle 300,300,500,500,red,red
circle 100,100,200,200,blue,blue
circle 600, 500,700,550,magenta,magenta

rlocate 400,200
rturn random(360)
while true // roam forever
      // forward until an object is found
      while rFeel()=0
            rForward 1
      wend
      // turn 180 degrees plus or minus 30 degrees
      rTurn 150 + random(60)
wend
End
```

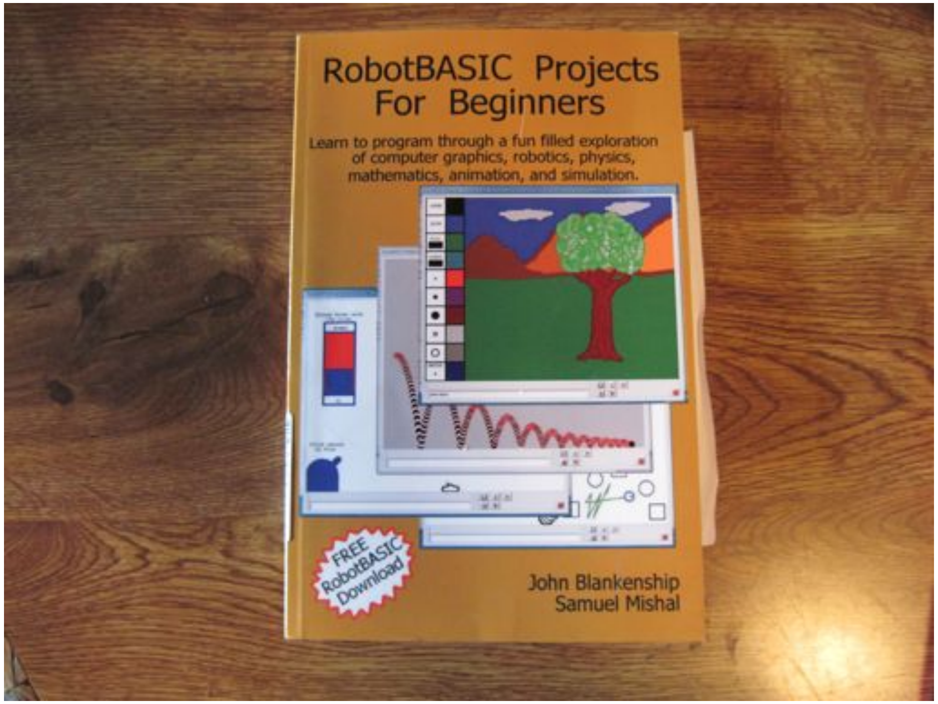
RobotBASIC IDE



RobotBASIC IDE V4.2.1 [Editor Screen]
File Edit Run Help
Ln#
autoBotTest.BAS
RobotBASIC Interactive Terminal Screen [Z:\home\john\autoBotTest.BAS]
Take Screenshot
Take Screenshot
○ Grab the whole desktop
○ Grab the current window
○ Select area to grab
Grab after a delay of 0 seconds
Effects
□ Include pointer
L:1 C:0 Ins Z:\home\john\autoBotTest.B
Menu
Sat Feb 19, 4:57 PM

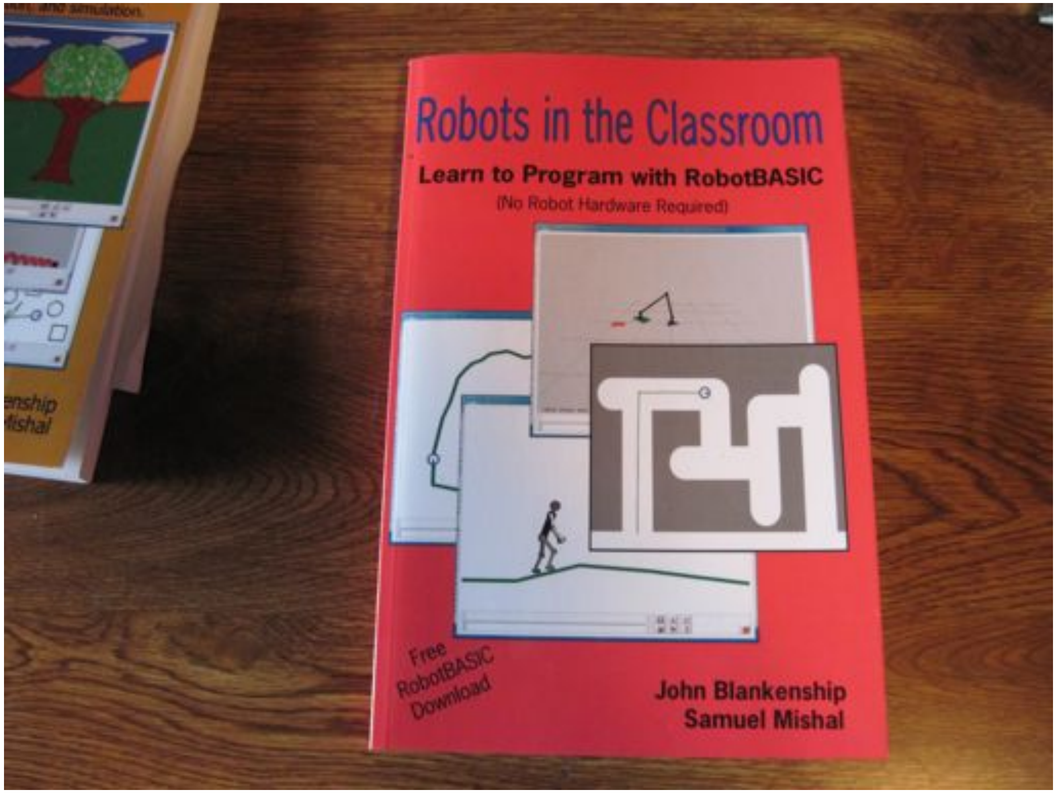
```
rectangle 300,300,500,500,red,red
circle 100,100,200,200,blue,blue
circle 600, 500,700,550,magenta,magenta

rlocate 400,200
rturn random(360)
while true // roam forever
	// forward until an obje
	while rFeel()=0
		rForward 1
	wend
	// turn 180 degrees plus
	rTurn 150 + random(60)
wend
End
```

RobotBASIC Projects for Beginners


RobotBASIC Projects
For Beginners
Learn to program through a fun filled exploration of computer graphics, robotics, physics, mathematics, animation, and simulation.
FREE RobotBASIC Download
John Blankenship
Samuel Mishal

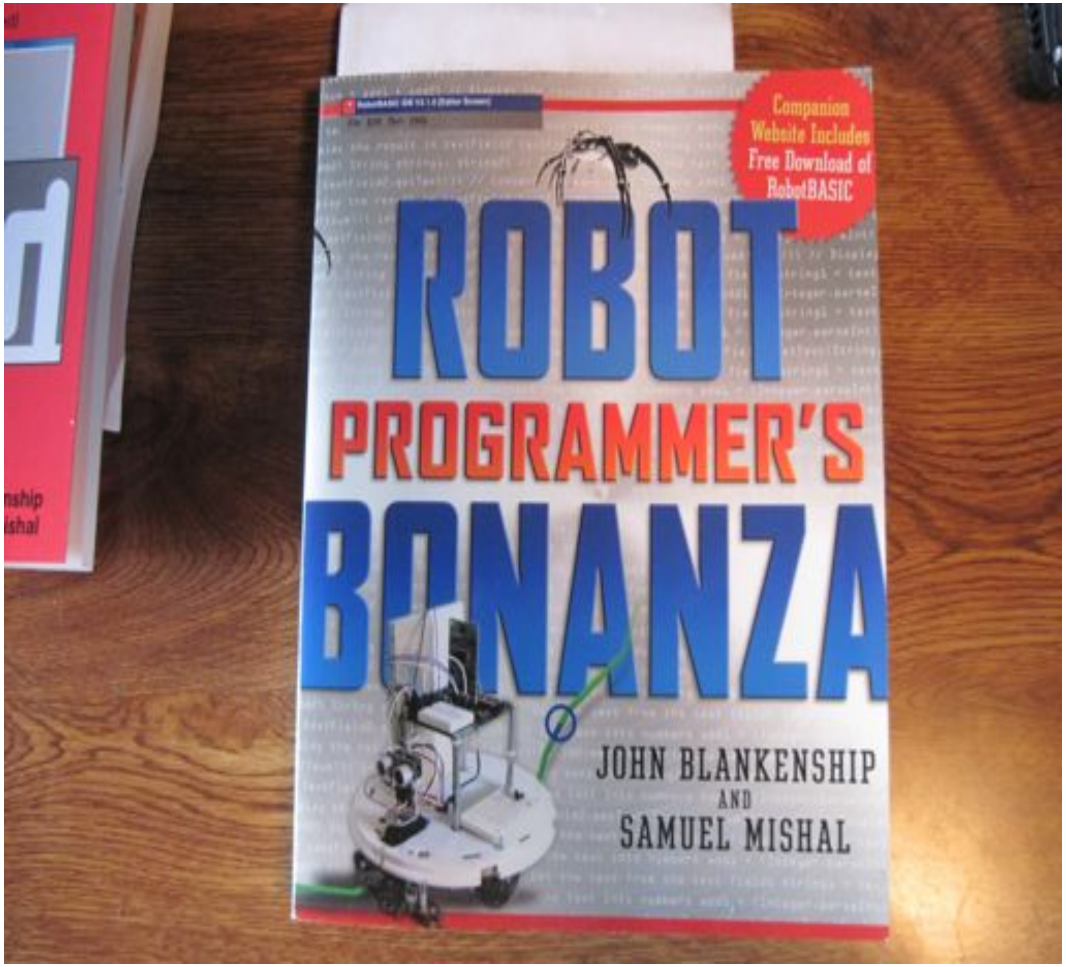
- Introduction to programming
- Full toolbox of control structures
- Modular programming
- No robotic hardware necessary

Robots in the Classroom



- Next Level Up
- No Robot hardware required
- Loops, variables, decisions, modules
- Line following, maze solving, beacon navigation
- Exercises

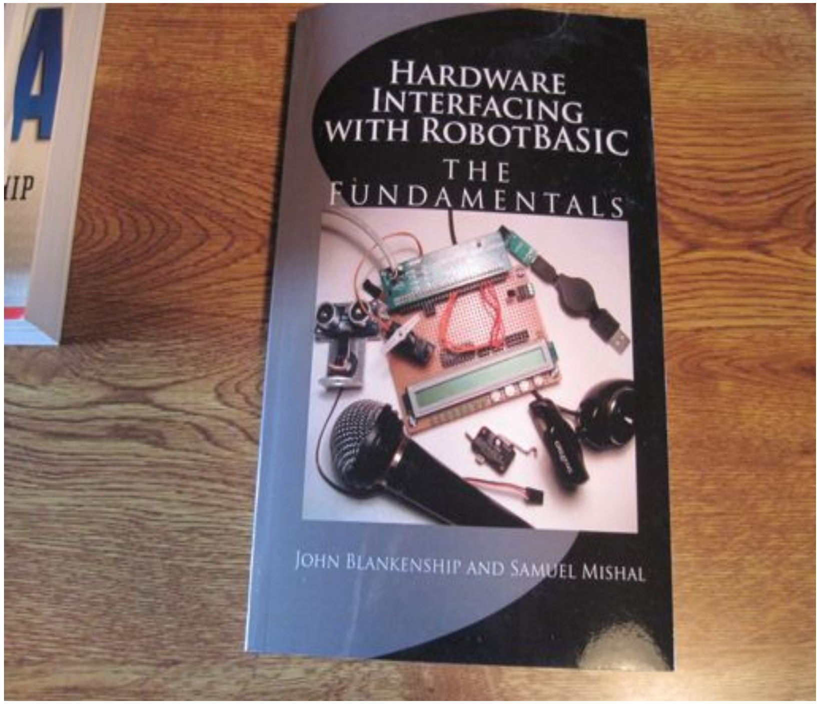
Robot Programmers Bonanza



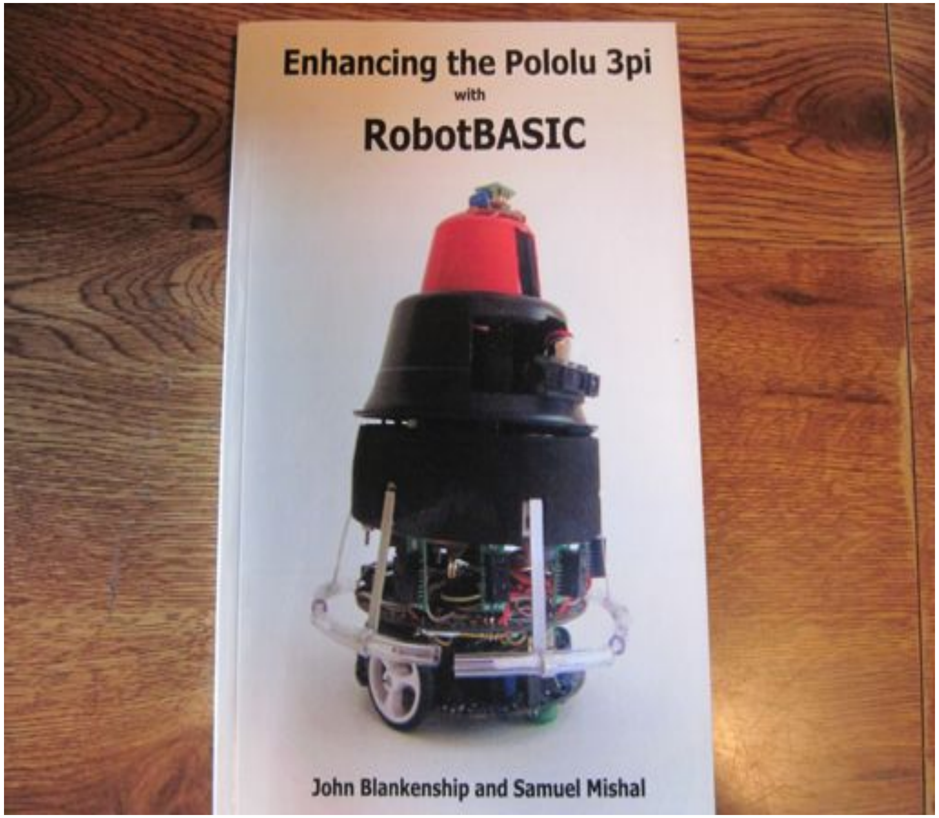
- The bible of RobotBASIC
- Robotic sensors
- RC control algorithms
- Random roaming
- Line following
- Wall following
- Avoiding drop offs
- Vector graphic robot
- Maze negotiating
- Maze learning
- Controlling a modified BoeBot with RoboBASIC

Hardware Interfacing with RobotBASIC

- Parallel port examples
- Serial port examples: BoeBot (Parallax Board of Education)
- Motor control: DC and servo (Pololu)
- Sensors: digital IR (Pololu), Ping sonar (Parallax), line sensor (Pololu), electronic compass (Parallax)
- Intro to sensor-expanded Pololu 3pi


HARDWARE INTERFACING WITH ROBOTBASIC
THE FUNDAMENTALS
JOHN BLANKENSHIP AND SAMUEL MISHAL

Enhancing the Pololu 3pi with RobotBASIC



- Advanced robotics
- Expanding sensor array on 3pi line-follower/maze-solver
- Some soldering required!
- Compiling virtual machine on 3pi using RobotBASIC code and Pololu libraries