

55 Stillpass Way
Monroe, OH
45050

J2R Scientific

(513) 759-4349
Weyoun7@aol.com

<http://www.J2RScientific.com>

How to Build a J2
by, Justin R. Ratliff
Date: 1/22/2007



The J2 robot (**Figure 1.**) from J2R Scientific is a compete robot ready to run out of the box. Its design is part of the REOP (Robotics Education and Out Reach program of The Robotics Club of Yahoo at <http://tech.groups.yahoo.com/group/theroboticsclub> or www.trcy.org

The J2 has been a fun project. My original idea was to create a security robot after the 9-11 attacks. After dedicated thought into all aspects of what a security robot would require, I decided it was better to start small so I could more easily model the behaviors and systems I would need for a security robot. This led me to create the J2 robot, which does bare a passing resemblance to a somewhat famous movie robot of the mid 80's.

Figure 1.

The J2 allowed me to experiment with Subsumption programming and the new PING sonar range modules from Parallax, inc.

In the creation process of J2 I decided I really needed to focus on what I wanted from my robot and what I thought other people would want to see. I decided my robot needed to look interesting; be easy to add components to; easy to remove components from; and be easy to use.

With these guidelines in mind I chose the Basic Stamp II as the micro-controller. I decided on a standard sensor compliment of Sonar for navigation, IR for close up or edge detection, CdS photo cell for light level and a multi-input port for multiple switch inputs from either a keypad or several bump sensors. I decided on two continuation rotation servos for drive motors and a servo controlled neck for the sonar to look up and down. The J2 did not need to turn the sonar side to side (I'll explain why). I wanted a speaker for tone generation, to give J2 a voice.

I also wanted to experiment with the new Text-to-Speech Emimic board from Parallax, inc. And have the ability to add more subsystems and components easily.

Once I knew what I needed to focus on I could design the board. **Figure 2** shows the J2 schematic and **figure 3** shows the PCB artwork as laid out using free software from www.expresspcb.com Prices for boards are very reasonable. One note of caution, double check the hole spacing and size before you place an order. With all the options it's easy to choose a wrong size or settings. It's a great program and service and I highly recommend them for anyone.

Figure 2.

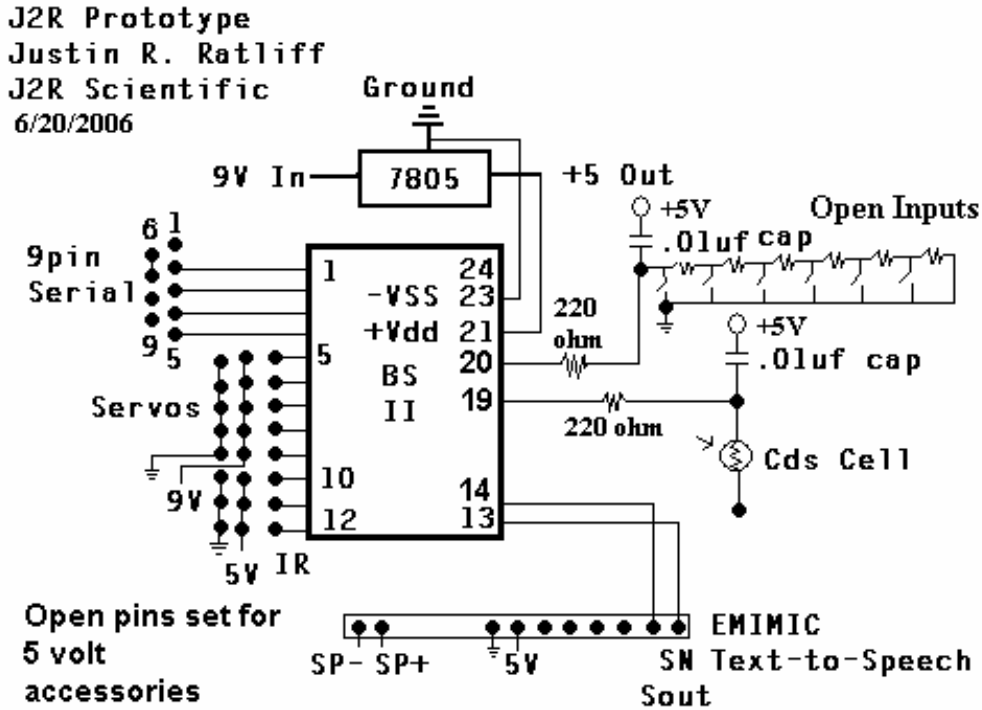
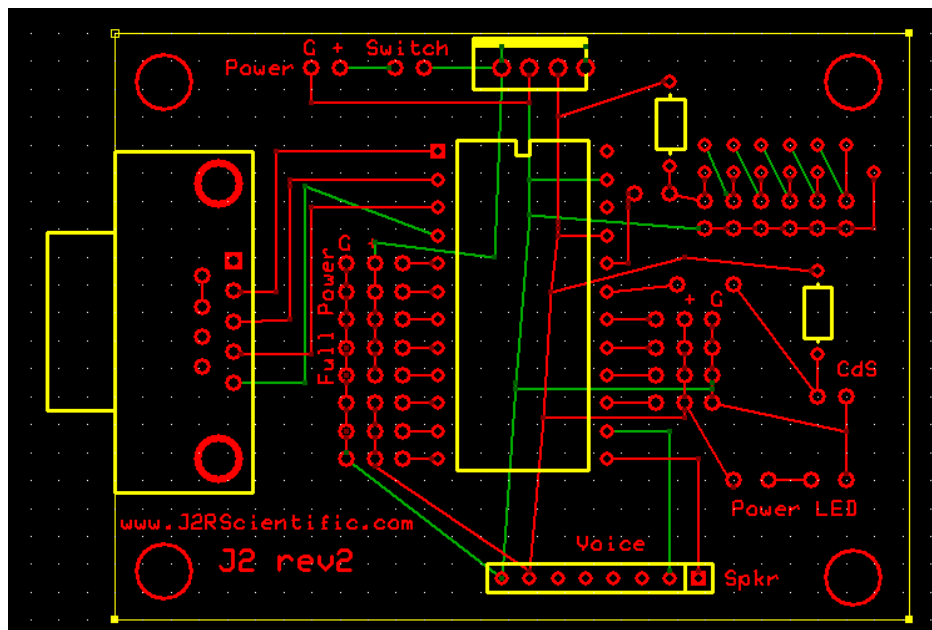


Figure 3.



Below is the listing for the standard i/o assignments on the J2:

Neck = i/o 2
Right Servo = i/o 1
Left Servo = i/o 3
GP2D12 IR = i/o 6
PING Sonar = i/o 7
Speaker = i/o 8
Text-to-Speech (if installed) = i/o 8 and 9
LED Mouth Option = i/o 10
CdS Option = i/o 14
Multi Input Option = i/o 15

Parts List:

Qty:	Parts:	Source:
1	Basic Stamp II	Parallax.com
2	Continuous Rotation Servos	Parallax.com
1	Standard Servo	Parallax.com
1	Emimic Text-to-Speech Module	Parallax.com
1	PING Sonar Module	Parallax.com
1	J2 PCB	J2R Scientific
1	J2 Body	J2R Scientific
1	9pin DSUB female connector	Digikey.com
1	24pin DIP Socket	Jameco.com
1	8pin SIP Socket	Jameco.com
2	Servo Wheels	Acroname.com
1	Sharp GP2D12 IR Module	Acroname.com
1	6 AA Battery Pack	
1	CdS Photo Cell	
1	Power Switch	
1	5V Regulator	
1	Caster Wheels	

Misc: wire ties, double sided tape, wire, heat shrink tubing, misc. connectors

The production J2 is built from stainless steel. The prototype body was built from Plexiglas. I cut the Plexiglas to size and heated it with a heat gun to bend it into shape. For a beginner I believe Plexiglas is cheaper, easier to work with and more forgiving than metal. It's also very light weight and fairly strong. **Figure 4** shows the prototype plastic J2 layout with larger foam wheels.

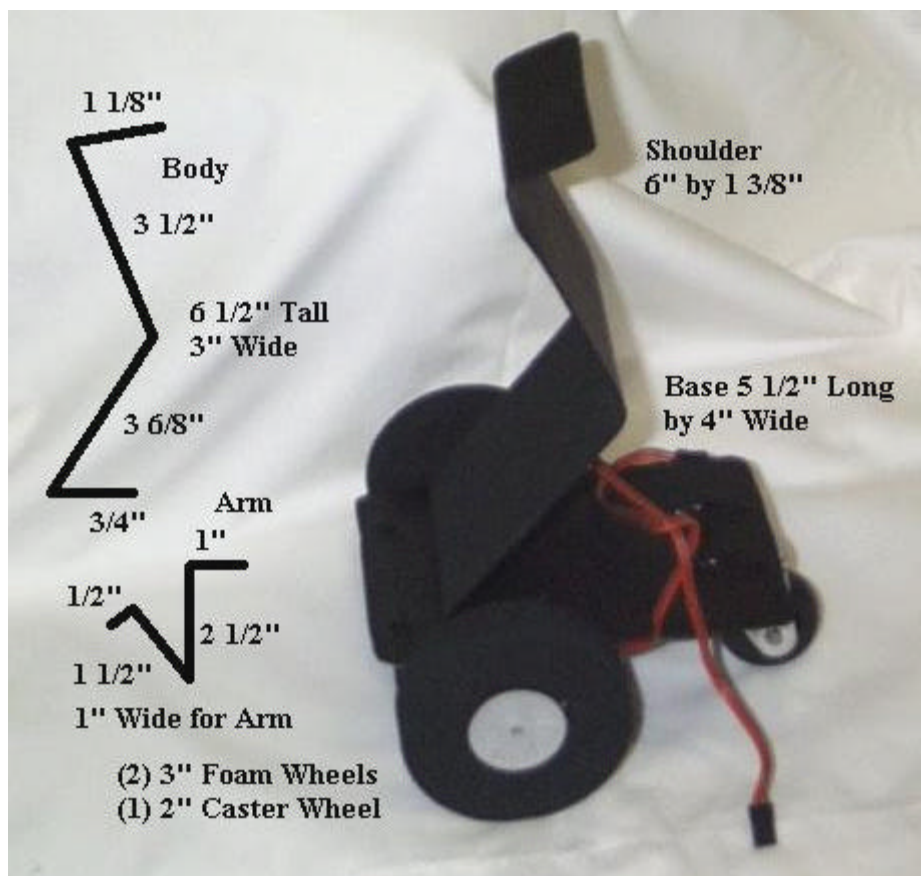


Figure 4.

A heat gun is sort of like an industrial hair dryer and you can pick one up at your local hardware store for around \$30. When heating plastic to bend it you want to heat the surface of the plastic where you'll place a bend and apply heat evenly. The plastic will need to get fairly hot, but if you see the surface structure start to change texture you know you've applied too much heat! With clear Plexiglas you'll see the surface start to granulize if it gets too hot.

If you choose to paint the Plexiglas I recommend you use a plastic primer spray paint from your hardware store to cover the Plexiglas before you apply your final paint color. A final layer of clear coat will add protection to your robot's paint and body.

With the J2 programs based on Subsumption behavioral subroutines it makes it easy to add new behaviors and adds at times at least the appearance of intelligence. To program the J2 you'll need to download the Basic Stamp editor either the DOS or Window's version from www.parallax.com. Below is the J2SubSumExplore program. You can download the latest J2 code from www.j2rscientific.com See **Figure 6** for the standard J2 Explore program.

Lesson Learned: It is always better to start small. Master all that you can, seek perfection from every part of your small robot before you move onto larger projects. Always apply the “keep it simple” approach to your design. Lay out your robot designs on paper with as much detail as possible in the hardware and programmed actions of the robot. Imagine your robot going through the actions of your program with the hardware you have drawn out. Many mistakes and limitations can be discovered before you begin building your robot if you invest the time to think through your design.

For instance, the J2 does not have a servo to turn the sonar side to side but it can look up and down, why is that? On such a robot (in my opinion) it would be wasteful to include a servo to rotate the head side to side when the robot can turn its wheels to look side to side. Also if the robot sees something of interest to the side, it will already be ready to drive toward what it sees. If a servo rotated the neck to the side, the robot would need to turn its body to match what the head saw.

This one detail of removing one servo from the head saves on power consumption, code space, and weight and makes the robot more efficient.

Figure 6. Robot Code

```
' {$STAMP BS2}
'J2SubSumExplore.BS2
'J2R Scientific
'12-12-2005
'J2 will explore about the world using subsumption based
'behavioral based intelligence.

'Generic values
tmp  VAR Word           ' tmp var, many routines
ltmp VAR tmp.BYTE0
htmp VAR tmp.BYTE1
seed VAR Word           'random number seed
val05 VAR Byte
time VAR Word
i    VAR Byte           'loop counter
a    VAR Byte
PING CON 7             'PING sonar port
LEFT CON 3             'left wheel port
RIGHT CON 1            'right wheel port
NECK CON 2             'neck servo port
SPKR CON 8             'speaker port
LED  CON 10            'LED mouth, this is not standard on all J2's

'These are for the servo routines
SACT CON 5             'times through act routine
drive VAR Word         'wheel command combo
ldrive VAR drive.BYTE1 'left wheel command
```

```
rdrive VAR drive.BYTE0 'right wheel command
aDur VAR Nib 'duration of pulsout
```

'normal list follows

```
rv CON $6432 'forward
fd CON $3264 'reverse
st CON $4d4d 'stop
tr CON $324d 'turn right
tl CON $4d64 'turn left
rl CON $6464 'rotate right
rr CON $3232 'rotate left
bl CON $644d 'backup turning left
```

'wander values

```
wDir VAR Word 'wander value
wDur VAR Byte 'wander duration
```

'avoid states and vars

```
avDir VAR Word 'direction
avDur VAR Nib 'duration
```

'bumper vars and constants

```
bumper VAR IN6 'bumper io pin
bstate VAR Nib 'bumper FSM state
bDir VAR Word 'bumper direction holder
bDur VAR Byte 'duration in that direction
```

'set up for running

```
wDur = 0 'clear wander duration
aDur = 0 'clear act loop counter
bDur = 1 'clear bumper duration, may need to change back to 0
bstate = 0
drive = st 'stop servo motors - not really needed
LOW LED
```

```
main: 'subsumption architecture
GOSUB wander 'random wander instinct is lowest priority
GOSUB avoid 'avoid running into stuff
GOSUB bumpck 'don't stay bumped into it = highest priority
GOSUB act 'acts on highest priority movement needed
' i.e. last to set direction

GOTO main
```

```
wander: 'randomly wander around
IF wDur > 0 THEN wDone1
RANDOM seed 'random direction
i = seed & %111 'mask off for 0-7 only
LOOKUP i,[fd,tl,fd,fd,fd,fd,tr,fd],wDir 'chose direction
seed = seed + i
wDur = (seed & %111111) + 20 'mask for 64 choices of duration
```

```

wDone1:
  wDur = wDur - 1    'decrement wander counter
  drive = wDir      'get direction
  PULSOUT NECK, 900
  LOW LED
  RETURN           'completed

act:
  'moves servo motors
  IF aDur > 0 THEN aDec 'already doing one, got here
  aDur = SACT          '# of main loops between pulseouts +1
  PULSOUT LEFT, ldrive * 10
  PULSOUT RIGHT, rdrive * 10
aDec: aDur = aDur - 1    'decrement act loop cntr
  RETURN

avoid:
  PULSOUT 7, 5
  PULSIN 7, 1, time
  time = time ** 2251
  IF time > 0062 THEN avdone

avfront:
  HIGH LED
  avDir = rl          'rotate away
  avDur = 15
  drive = rl
  GOTO avdone

avdone:
  RETURN

bumpck:
  IF bumper = 0 THEN bmpnow
  IF bDur > 0 THEN bmpact

  BRANCH bstate, [bDone1, bbup]

breset:
  'end state 2, now reset
  bstate = 0         'state machine to idle
  RETURN

bbup:
  'end state 1, now
  bDir = rl         'rotate left away
  bDur = 65        'sets time limit
  bstate = 2       'next state
  GOTO bdrive

```

bmpnow: 'being bumped now
 bDir = rv 'set backup while bumped and
 bDur = 61 'for a while (+1) after not being bumped
 bstate = 1 'start state machine

bmpact: 'bump mode active
 bDur = bDur - 1 'decrement bump timer

bdrive:
 drive = bDir 'set drive direction to bump

bDone1: 'no bump
 RETURN

END