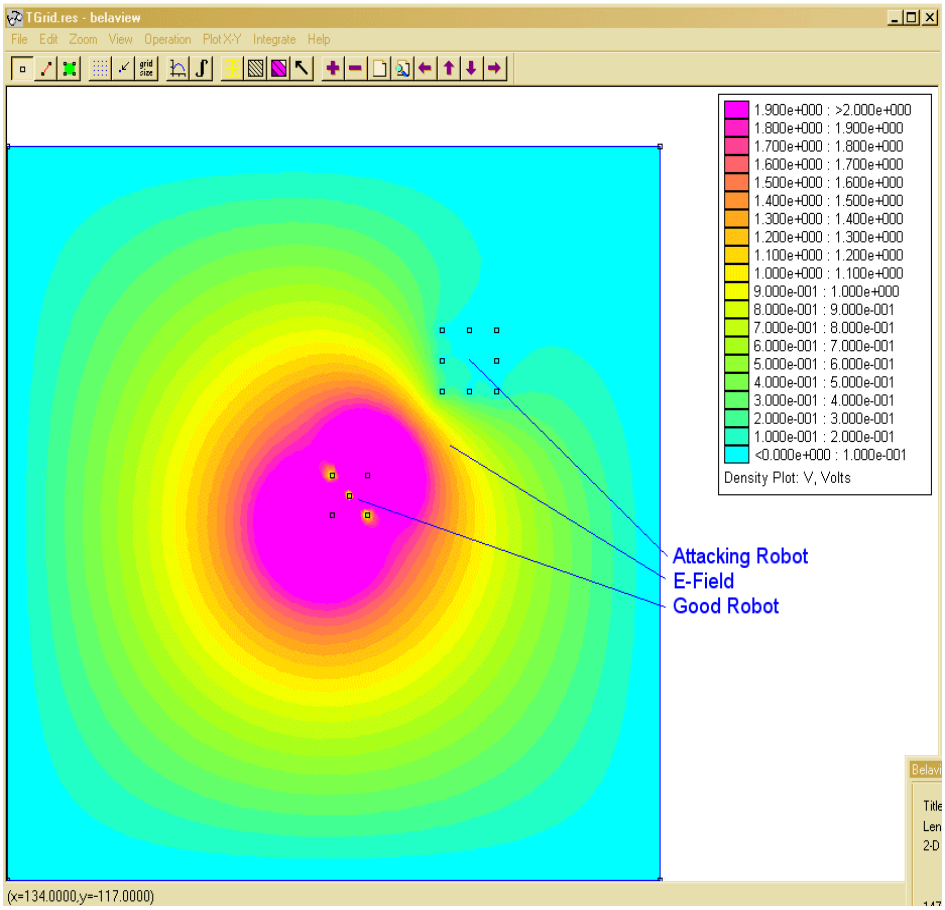
ThereminVision-II

Instruction Manual

Version 1.02
Terry Fritz



www.thereminvision.com

©TWF Power Electronics 2004 – Terry Fritz

Index

Index..... 2
Introduction..... 3
Theory of Operation..... 5
How to Use..... 8
Connections..... 9
Antennas..... 13
Adjustment..... 15
Example program..... 16
Example robot..... 17
Specifications..... 18
Schematic Diagrams..... 19
Parts List..... 21
Component Placement..... 22
Assembly Instructions..... 23
Trouble shooting..... 32
References..... 33

Introduction

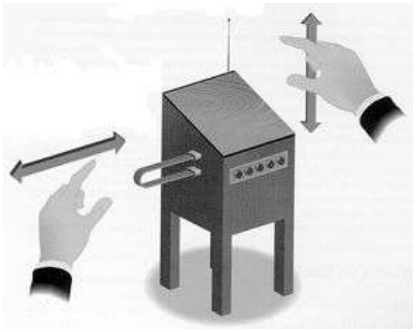
ThereminVision is a new robotic sensing system based on a very old idea. The “theremin” was the World's first electronic musical instrument invented in 1919 by Leon Theremin. The theremin uses two extremely sensitive antennas whose capacitance is varied by placing one's hands nearer or further from the antennas. The small capacitance variation controls oscillators that detect the tiny variations in capacitance due to objects moving near the antenna. In the case of the theremin, one antenna controls the volume while the other antenna controls the pitch of the instrument's sound. The theremin is easily capable of detecting objects many feet away. As Leon Theremin stated in his patents, such a system can be used in many ways and the number of antennas can be increased as needed. In our case, the principle of the theremin has been adapted to robotic sensing.



Leon Theremin



Modern Moog Theremin



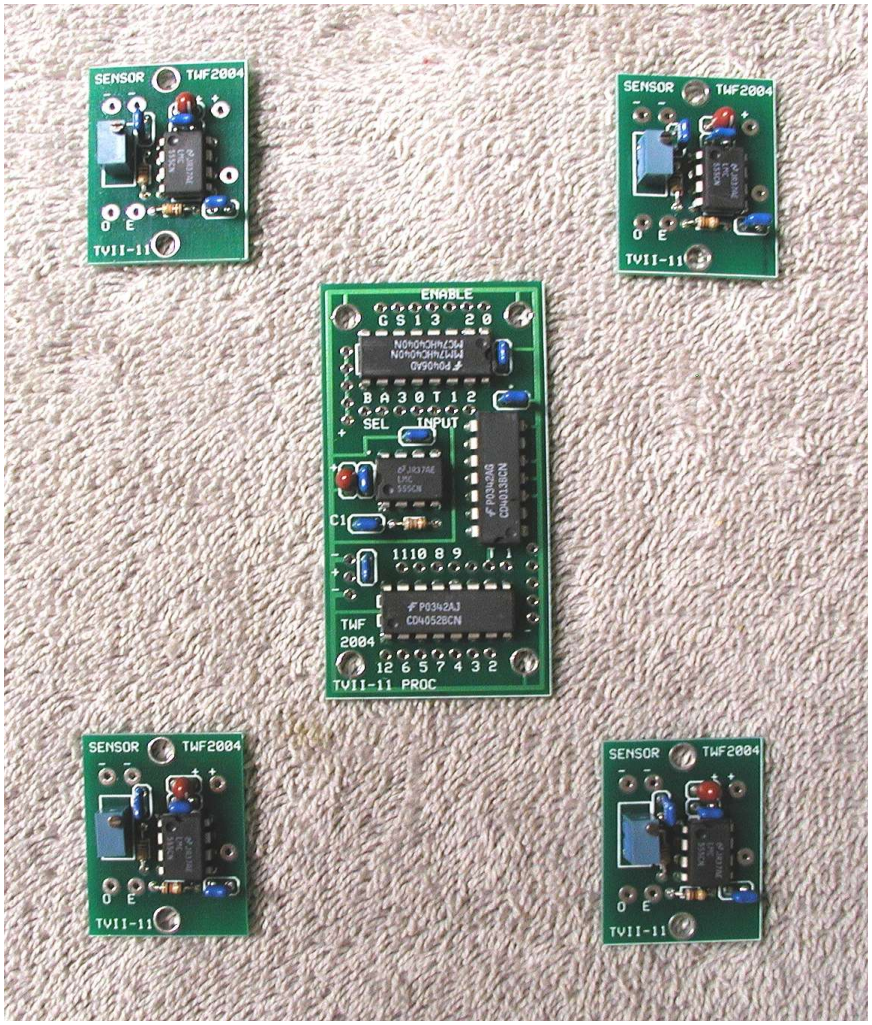
Hand movements for control

ThereminVision is a modernized and simplified version of the theremin with all digital circuits. The system can very quickly scan the area around and detect capacitance changes on the antennas around a robot. Each of four antennas can be selected by a microcontrollor and the resulting relative antenna capacitance is returned to the processor in the form of a variable pulse width. As an object nears a given antenna, the increasing capacitance causes the pulse width of the signal to the processor to decrease and visa-versa.

The sensitivity of the system is on the order of 0.001 pico Farad. Any object with a relative dielectric constant differing from “1” will be detected. Since practically all solids have a considerable relative dielectric constant, it is best to list the few things such a system may not detect. These would include air, vacuum, most gases. One solid that the system is relatively insensitive too is low density Styrofoam which can have a relative dielectric constant as low as 1.01. Of course, materials such as metal, human flesh, plastics, wood, and other common construction materials are easy for such a system to detect. Cardboard, wood,

and many porous organic materials also may have a fairly high water content which is easily detectable. The system is unaffected by light, sound, radio waves, and magnetic fields.

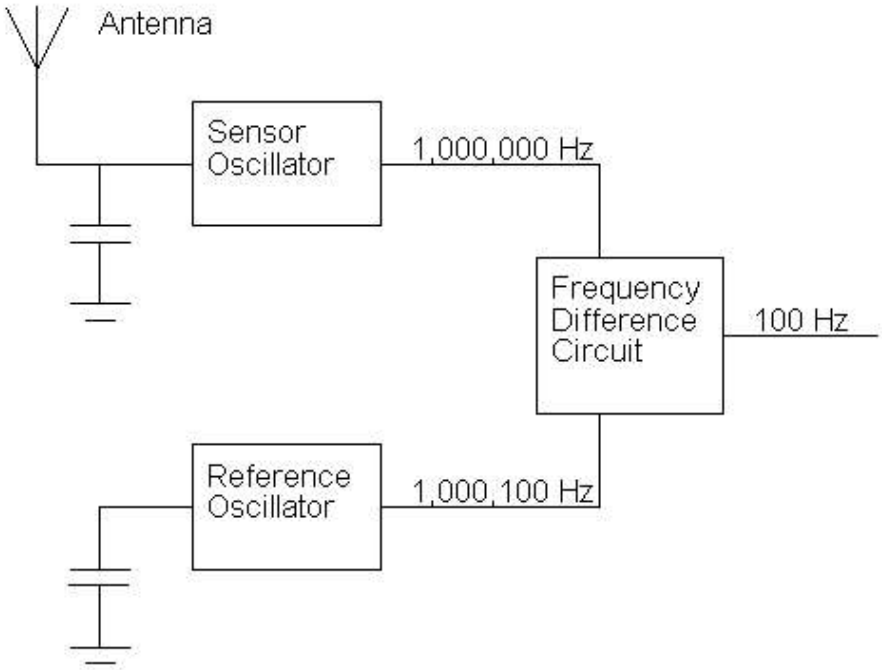
The system uses very low energy oscillators which are dependent on the capacitance of the antennas. The signal on a selected antenna is a 1.7 volt peak to peak triangle wave at approximately 1.7 MHz. The running ThereminVision system uses less than 2 mA of current at 5 volts DC. The system is very stable and easily adapted to today's many “pic” style microcontrollors commonly used in small robots.



ThereminVision-II Processor and four sensors.

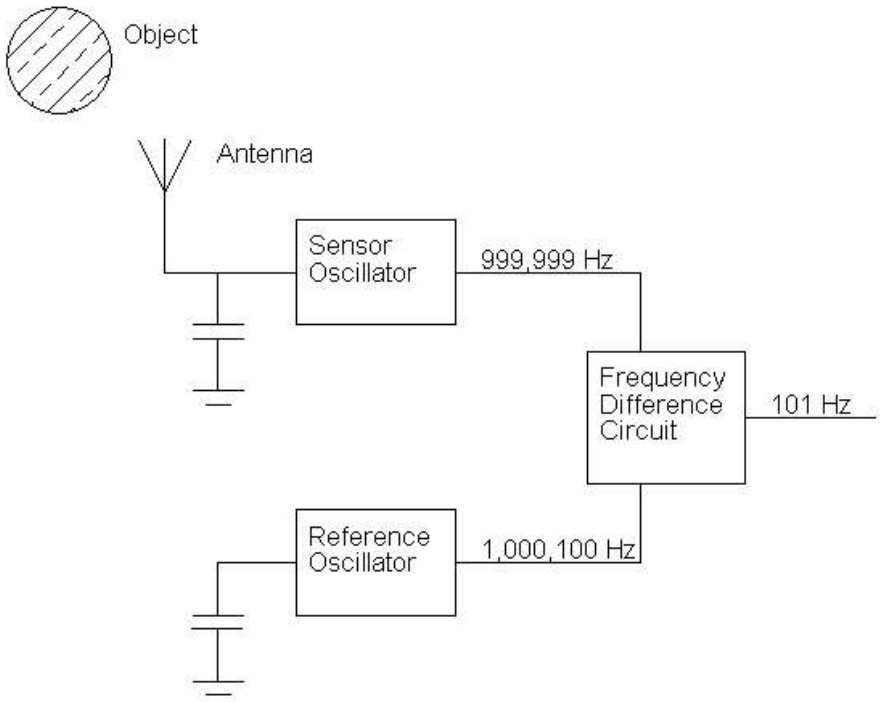
Theory of operation

The principle of ThereminVision is not complex. A sensing oscillator is setup with an antenna serving as a capacitance determining the oscillator's frequency. A second very similar oscillator is setup at near the same frequency but with a stable fixed capacitor and is used to provide a non-variable reference frequency. The output of these two oscillators is combined into a signal whose frequency is the “difference” of the sensing and reference oscillators. For example, if the sensing oscillator's frequency is 1,000,000 Hz and the reference signal's frequency is 1,000,100 Hz, the difference signal would be 100 Hz.



No object case.

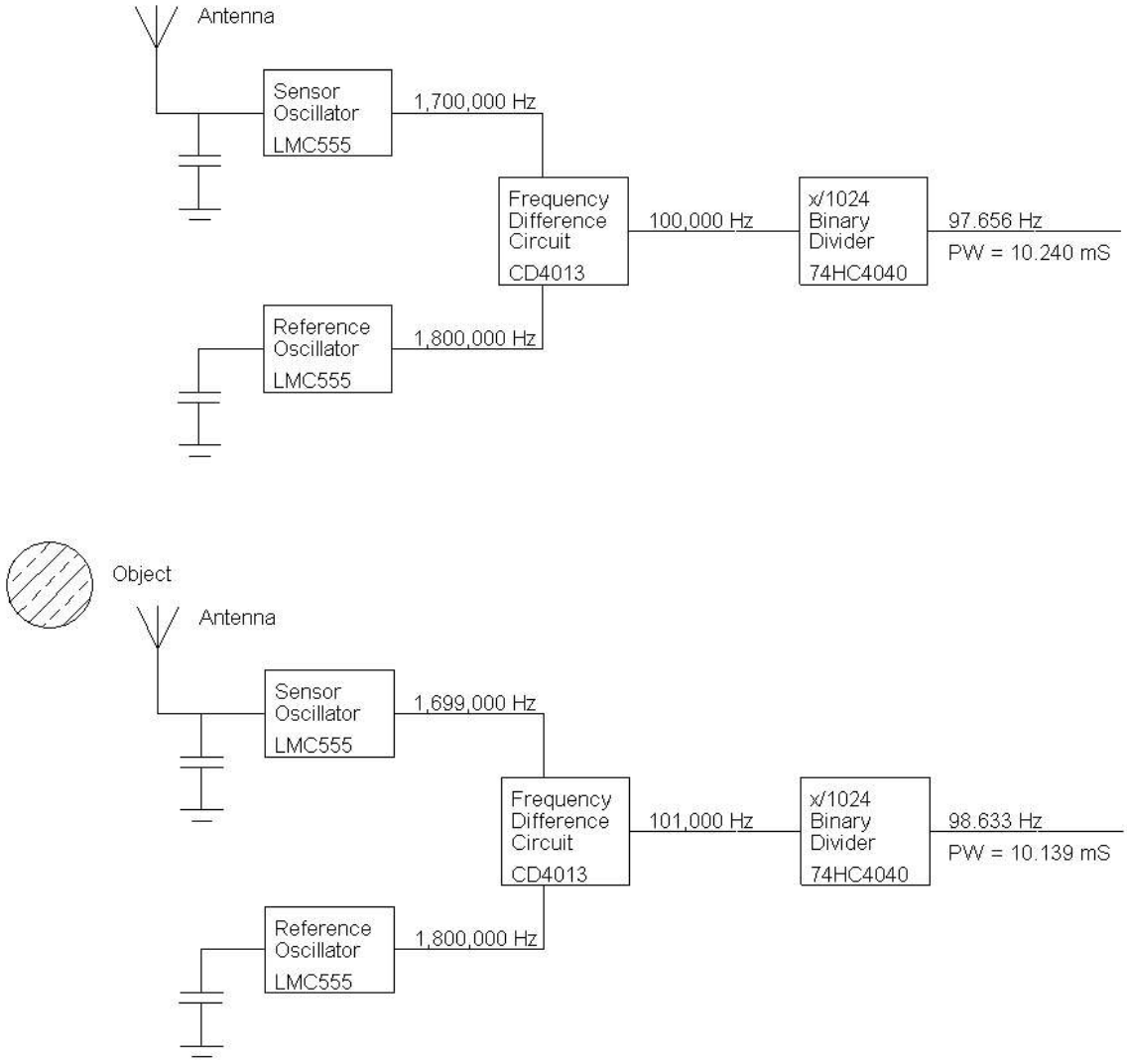
Now, suppose an object nears the sensing oscillator's antenna and changes the frequency to 999999 Hz. The difference signal is then 101 Hz. Thus, a 1 part in a million change in antenna's capacitance changes the output by 1%. This is a increase in sensitivity of 10,000 X which is the key to the remarkable sensitivity of ThereminVision to objects near the antenna.



Nearby object slightly lowers sensor oscillator's frequency.

In real terms, if the sensing antenna is 10 pF, we could detect a change in antenna capacitance of only 1/1000th of a pF with the above system. The ~1MHz sensing and reference signals are highly stable providing a reliable reading.

In practice, the ThereminVision system uses approximately 1.8MHz for the reference oscillator's frequency and 1.7MHz for the sensor's frequency for a sensitivity gain of about 18X. This gives a difference frequency of about 100 kHz. A digital counter is then used to average and add that frequency say 1024 times to give a sensitivity gain of 18 x 1024 or 18432X and reduce the frequency to 100000 / 1024 or ~100 Hz. The advantage of using a large difference frequency is it gives a very wide stable capacitance range for robotic applications. This allows distant sensing as well as sensing objects very close to or even touching the antennas. The digital counter vastly reduces “jitter” and reduces the frequency to the range of ~100 Hz which is easily in the range for a processor to accept and measure with high accuracy. The exact frequency and divider values used are not critical at all and can be easily adjusted as needed. The ThereminVision system's antennas are in parallel with a 22pF capacitor as well as the capacitance of nearby parts of the robot itself. Small potentiometers on each sensor allow easy adjustment in any case.



Typical ThereminVision-II system.

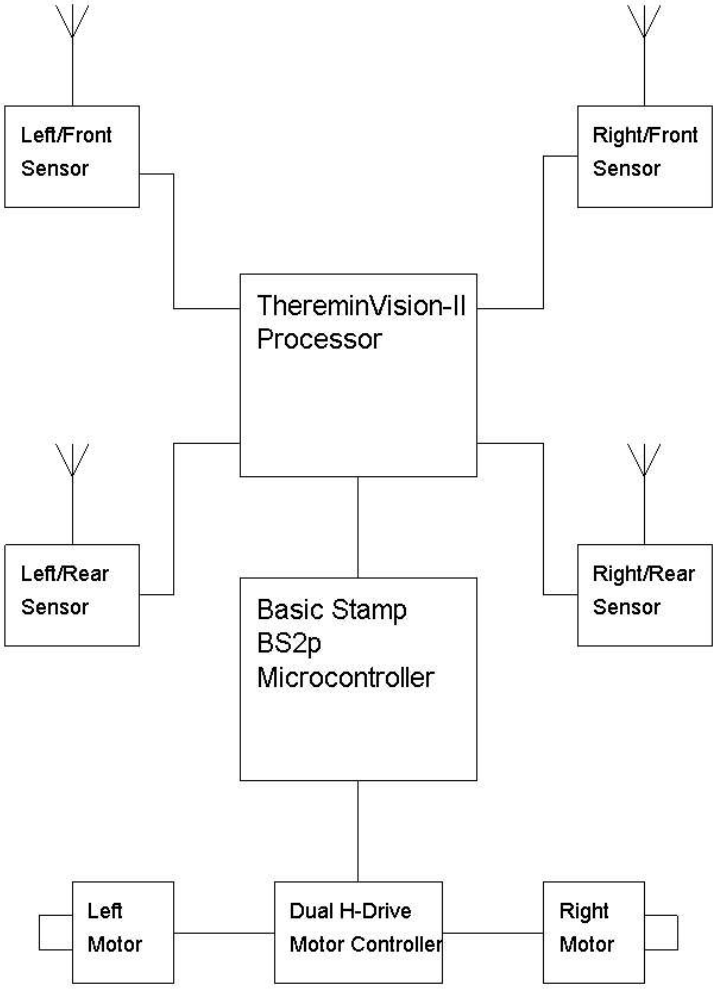
If we use a microcontroller capable of 1 uS resolution and thus it detects a positive pulse width as 5000 counts for our 100 Hz signal, we can detect a variation of 1 part in 5000 of the 100 kHz difference signal. This represents a variation of 1 part in 5000 x 18 (or 1/ 90000 variation) of the sensor's signal frequency. Since we have ~25 pF as antenna capacitance normally. We can detect 1 part in 90000 of this in variation. That works out to an over all sensitivity of 25 / 90000 or 0.00028 pF!

How to Use ThereminVision-II

The ThereminVision-II system is composed of a processor that controls four sensors which are typically located at the four corners of a robot.

The processor provides +5 volts, ground, and enable signals to each processor and the signals are returned back to the processor. Each sensor is connected to the processor via four wires for these signals.

The processor is supplied with +5 volts, ground, and sensor select signals from a microcontroller. The processor supplies an output signal back to the microcontroller. The microcontroller typically selects a sensor via the sensor select lines and the reading is returned to the microcontroller as a pulse width on the output signal line.



Typical ThereminVision-II robot control system.

Connections

Connection for the ThereminVision-II system are as follows referring to the following system connection diagrams.

Processor

+5 Volt Power – Power to the ThereminVision-II system. 5 VDC @ 2mA.
Ground – Power ground to the ThereminVision-II system.

Sensor +5V Power – Four power connections for each of the four sensors.
Sensor Ground – Four power ground connections for each of the four sensors.

Sensor Select A – Bit 0 from microcontroller to select the sensor being used.
Sensor Select B – Bit 1 from microcontroller to select the sensor being used.
Sensor Stop – A HIGH signal turns off all sensors. May be grounded if not used.

Select A	Select B	Stop	Sensor 0	Sensor 1	Sensor 2	Sensor 3
LOW	LOW	LOW	ON	OFF	OFF	OFF
HIGH	LOW	LOW	OFF	ON	OFF	OFF
LOW	HIGH	LOW	OFF	OFF	ON	OFF
HIGH	HIGH	LOW	OFF	OFF	OFF	ON
X	X	HIGH	OFF	OFF	OFF	OFF

Sensor Enable 0 to 3 – Individual enable signals to each sensor.

Sensor Input 0 to 3 – Individual frequency input signals from each sensor.

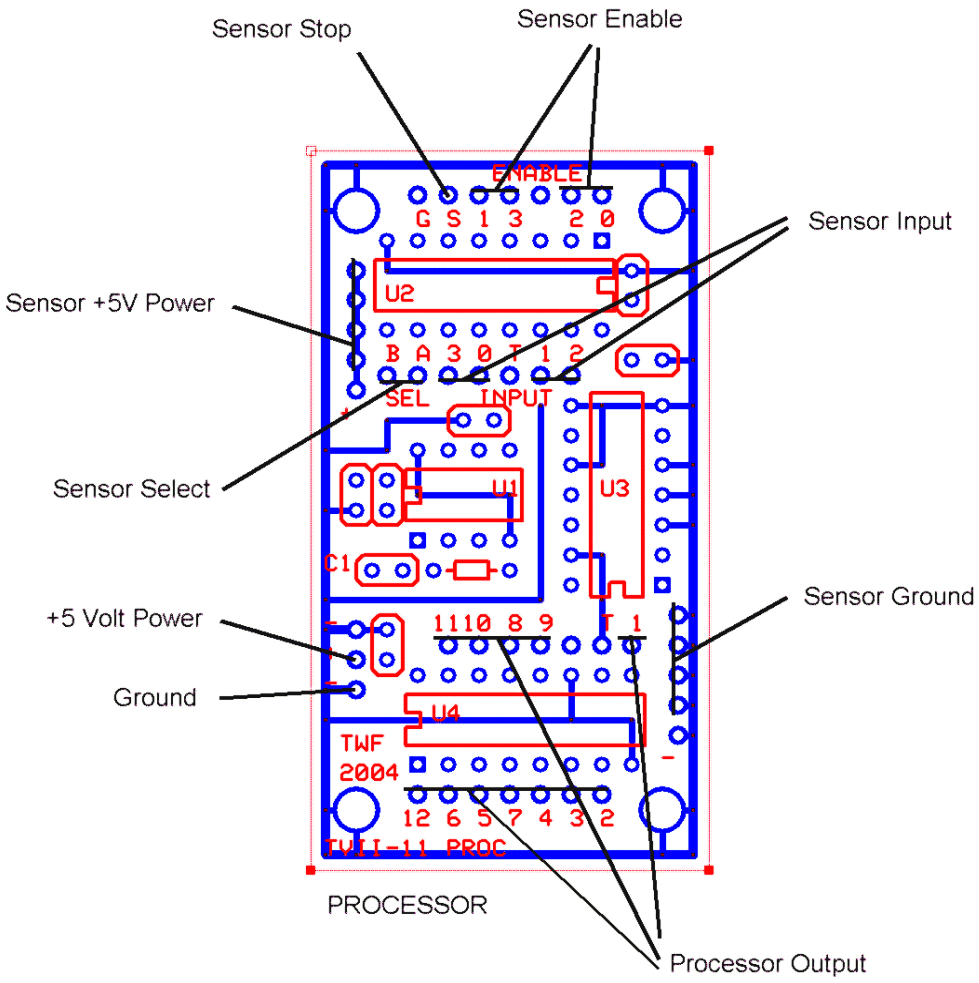
Processor Output 1 to12 – Pulse width output signals to the microcontroller. The divisor is 2^n. Thus, output 2 divides by 4, output 3 divides by 8,...

Sensors

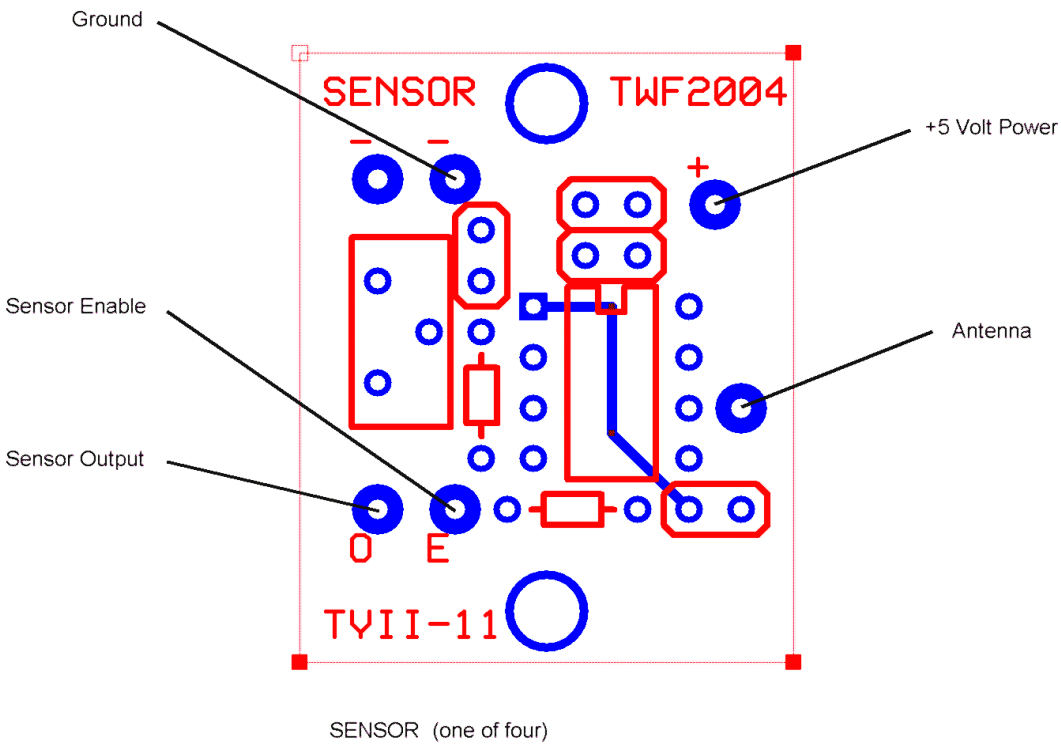
+5 Volt Power - +5 VDC power from the processor.
Ground – Ground from the processor.

Sensor Enable – Enable signal from the processor to turn on the sensor.
Sensor Output – Frequency output signal to the processor.

Antenna – Connects to the sensor antenna.

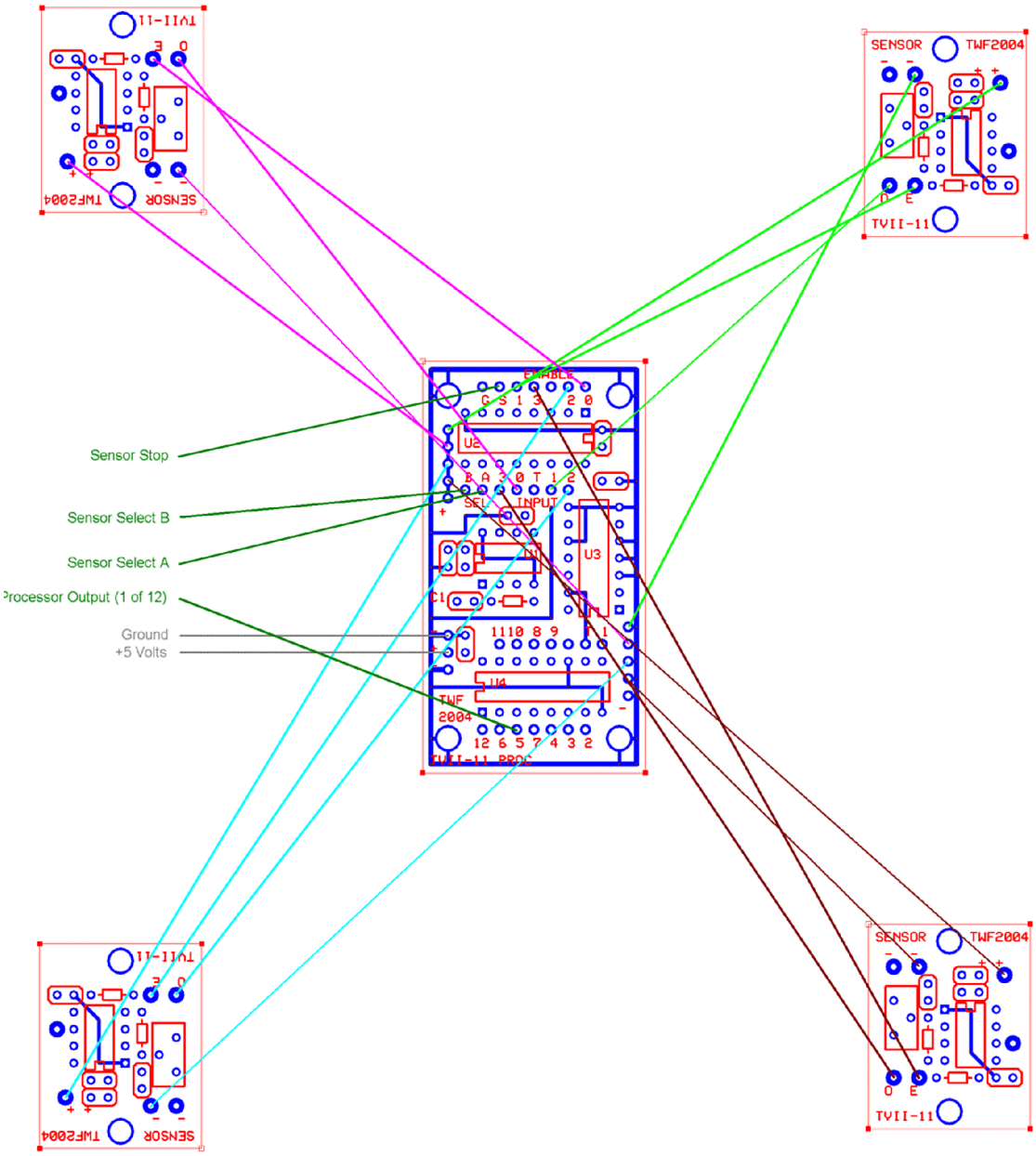


ThereminVision-II Processor Connection Diagram.



SENSOR (one of four)

ThereminVision-II Sensor Connection Diagram.



ThereminVision-II System Connection Diagram.

Antennas

ThereminVision-II is designed to work with four corner antennas that detect objects near each of the four corners of a robot. These antennas need to be rigid and objects near them need to be fixed in place. Otherwise, movement of the antennas will cause false readings. It does not matter if metal objects are near the antenna as long as they are rigid since the fixed objects will not change the readings over time. The software or adjustment pots can be used to compensate for the fixed objects.

The antennas can also detect platform edges. As the antenna nears and edge, the pulse width will increase. This can be used for edge detection.

For the antenna to be most sensitive to an object, the antenna should be designed to expose as much surface area as possible toward an expected object. The capacitance between the antenna and an object is generally given by the equation:

$$C = k A / d$$

Where:

C = The capacitance between the antenna and the object (Farads).
k = A constant (8.854 Farad / meter).
A = Area between the two objects (square meters).
d = Distance between the objects (meters).

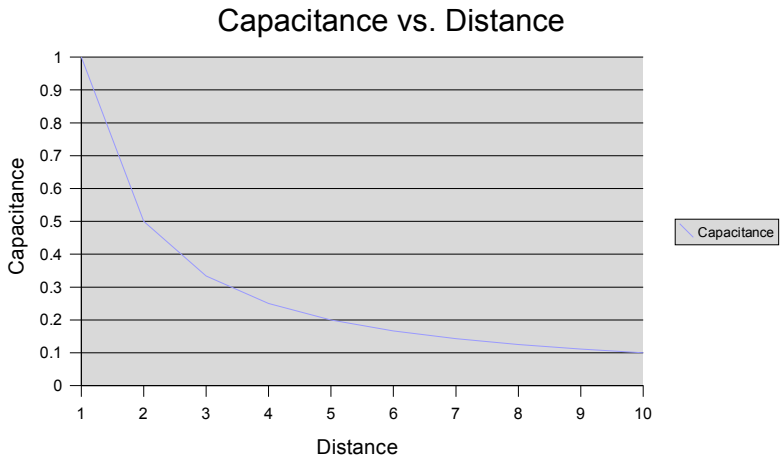
Although this specific equation is not really needed for our work, a few ideas from it should be understood.

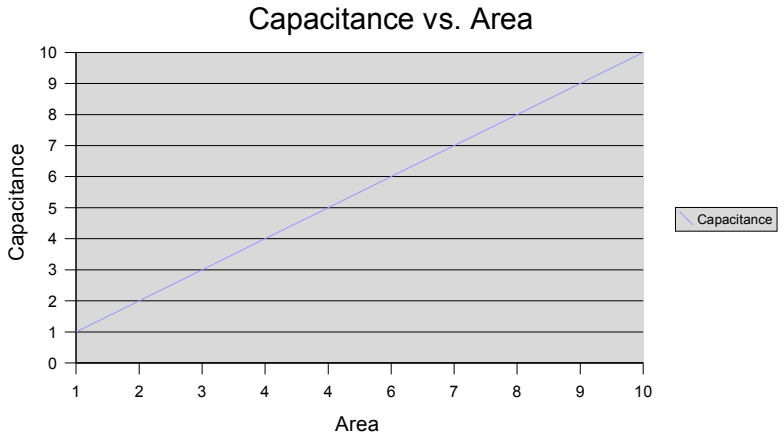
1. The capacitance will increase in proportion to 1 / distance. Thus, on object 2 feet away will have 1/4 the capacitance of on object 1 foot away.

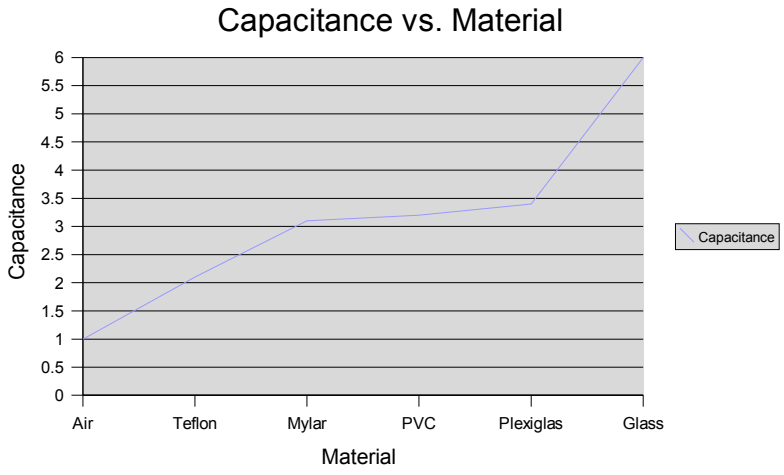
2. The capacitance is proportional to the area between the objects. Thus, a 4 square inch object will have four times the capacitance of a 1 square inch object.

3. Metals and conductive objects will show up more easily than say light plastics. However, most materials will be detected.

Below are some charts demonstrating these effects:







Effects of various antenna parameters on object detection capacitance.

Adjustment

The sensors can be adjusted over a wide range to fit many situations. There is no “one way” to adjust them, but the following works well.

The reference oscillator on the processor runs at about 1800kHz. Thus each of the sensor can be adjusted to about 100kHz lower or 1700kHz. This allows the pulse width to decease over a wide range. As objects come near, they drop the sensor frequency further. The closer a sensor's frequency is to the reference oscillator, the more sensitive it is. If they are too sensitive, then the readings will be unstable. In general, the sensors should not be adjusted to a higher frequency than the reference frequency where the pulse width increases as an object comes near.

The pulse width of 100kHz is 5uS. This can be divided by say 1024 by pin 14 of U4 for about a 5mS pulse width. A microcontroller can be used to check this. Of course, if an oscilloscope or frequency counter is available then it is trivial. One could also just adjust it by trial and error, but it is best if the sensors are at least adjusted to about the same frequency so objects will give similar readings on each sensor. In many cases, a simple test program can be incorporated into the microcontroller program to aid in the sensor adjustments. Below are the equations to find the pulse width and the resulting microcontroller readings.

Pulse Width = Divider / (2 x Difference Frequency)

Result = Divider / (2 x Difference Frequency x Resolution)

Difference Frequency	Divider	Pulse Width	Processor Resolution	Result
100kHz	1024	5.12mS	0.75uS	6827
100kHz	512	2.56mS	2.0uS	1028
250kHz	4096	8.192mS	0.75uS	10923

The pulse width can be adjusted with the sensor pots and by which divider ratio is selected on the U4 binary divider IC. The smaller the difference frequency set with the pots on the sensor is, the more sensitive it becomes. Higher divider ratios allow higher resolution to the microcontroller.

When adjusting the pots, one's hand and the tool will affect the readings. Thus, you make a little adjustment and take your hand away to check it.

Example Program

This program for the Basic Stamp BS2p microcontroller reads the raw data from the four sensors and displays it to the debug screen.

```
'{$STAMP BS2p}
'{$PBASIC 2.5}
'This example program runs the ThereminVision-II and displays raw output
numbers.
'pin0 of the Basic Stamp BS2p is connected to the ThereminVision processor
sensor stop (PIN 6 of U2).
'pin1 is connected to select A (pin 10 of U2).
'pin2 is connected to select B (pin 9 of U2).
'pin4 is connected to the processor divide by 4096 output (pin 1 of U4).

'define variables
sensor0 VAR Word                        'sensor input variables
sensor1 VAR Word
sensor2 VAR Word
sensor3 VAR Word
stoppin PIN 0                           '0 = sensors on, 1 = sensors off
selecta PIN 1                           'sensor select A
selectb PIN 2                           'sensor select B
DIRA = 7                                'set pins 1 - 3 for output
signalin PIN 4                          'pin 4 is the pulse input
DEBUG CLS                               'clear output screen

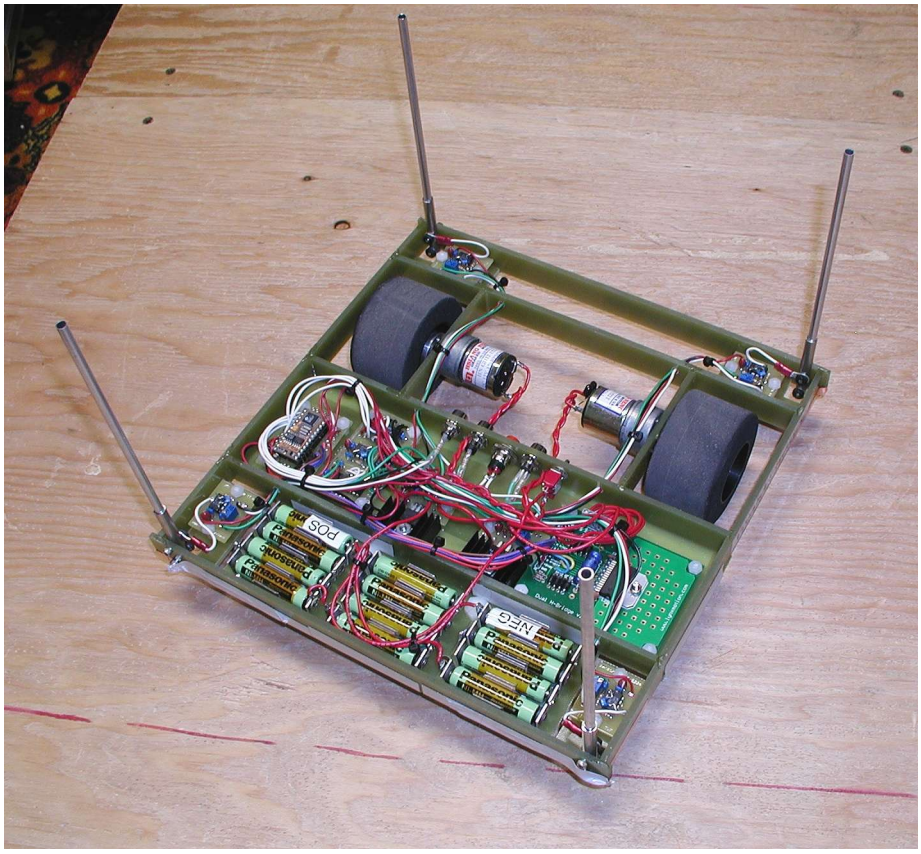

start:
GOSUB scan                              'scan the four sensor readings
GOSUB display                           'display the readings on the LCD
GOTO start                              'repeat loop


scan:
'Scan the four sensors and store the pulse widths
selecta = 0 : selectb = 0               'select sensor 0
PAUSE 1                                 'allow sensor to stabilize
PULSIN signalin, 1, sensor0             'get pulse width
selecta = 1 : selectb = 0               'select sensor 1
PAUSE 1                                 'allow sensor to stabilize
PULSIN signalin, 1, sensor1             'get pulse width
selecta = 0 : selectb = 1               'select sensor 2
PAUSE 1                                 'allow sensor to stabilize
PULSIN signalin, 1, sensor2             'get pulse width
selecta = 1 : selectb = 1               'select sensor 3
PAUSE 1                                 'allow sensor to stabilize
PULSIN signalin, 1, sensor3             'get pulse width
RETURN

display:                                'display sensor data to screen
 DEBUG CRSRXY,20,7,SDEC5 sensor0,"  "
 DEBUG CRSRXY,10,7,SDEC5 sensor1,"  "
 DEBUG CRSRXY,10,2,SDEC5 sensor2,"  "
 DEBUG CRSRXY,20,2,SDEC5 sensor3,"  "
RETURN
```

Example Robot

The robot “FieldTheory” was designed to use ThereminVision for autonomous control. It is a 2 pound Critter Crunch class robot. It has ThereminVision sensors at each conner that connect to the processor. The processor connects to a Basic Stamp BS2p microcontroller. The microcontroller is connected to two drive motors through a H-bridge. The Basic Stamp program is given at the end of the reference section. The program does on-the-fly calibration, auto zeroing, and has a section to readout parameters if a computer is connected to the robot. The robot can detect the platform edge by looking for an increase in the sensor pulse width. It detects opponent robots by looking for a decrease in pulse width. The microcontroller directs the robots motion as appropriate for each and allows it to wander if nothing is detected.



The Basic Stamp BS2p program that runs FieldTheory is at the end of this manual.

Specifications

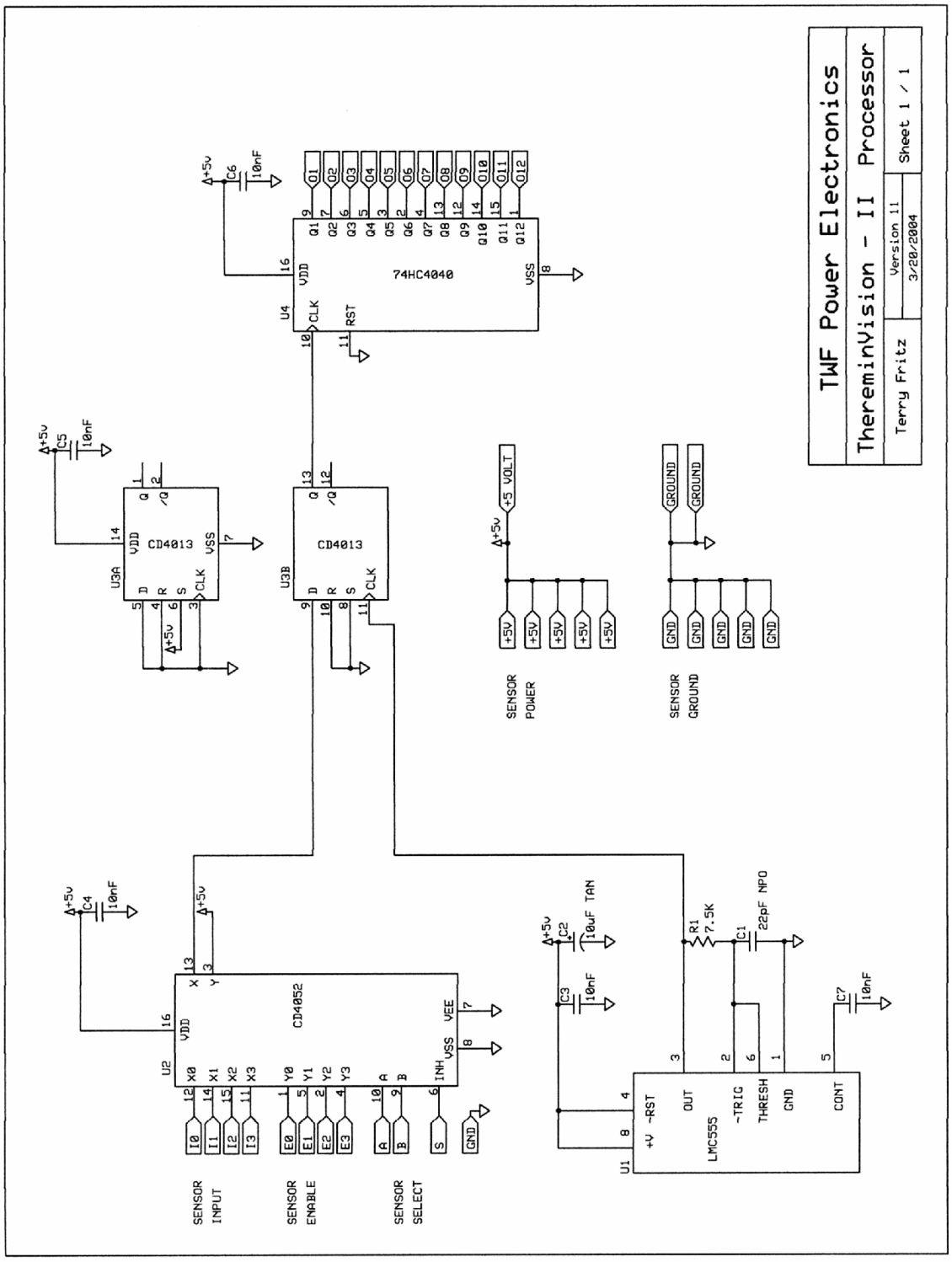
Processor

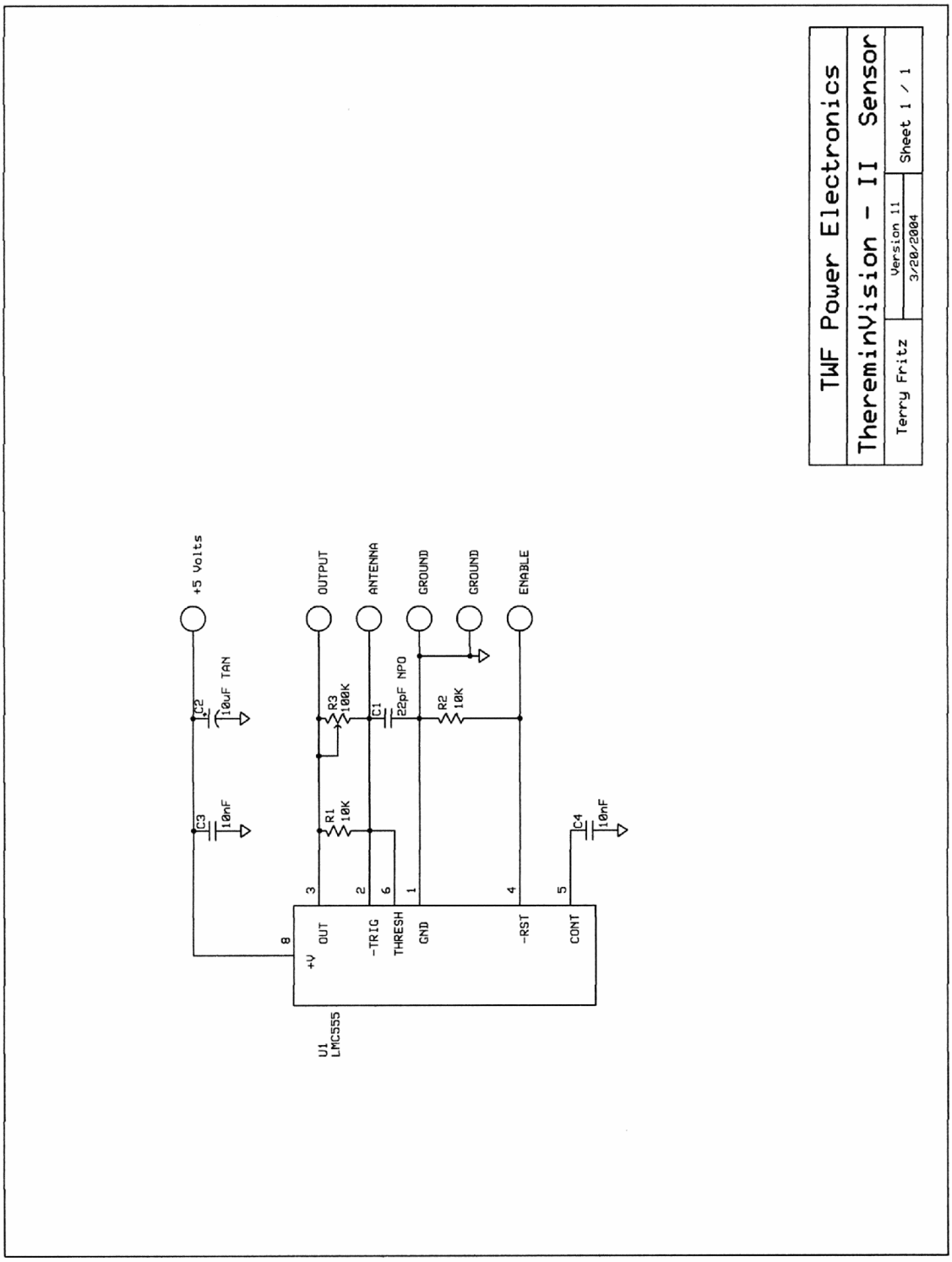
Dimensions	1.3 x 2.4 x 0.38 inches
Weight	10.7 gram
Power	900uA @ +5 VDC
Mounting	#4 screw holes on 1.0 x 2.0 inch centers

Sensor (each)

Dimensions	0.95 x 1.20x 0.5 inches
Weight	3.7 gram
Power	100uA disabled 700uA enabled @ +5 VDC
Mounting	#4 screw holes on 1.0 inch centers

Schematic Diagrams

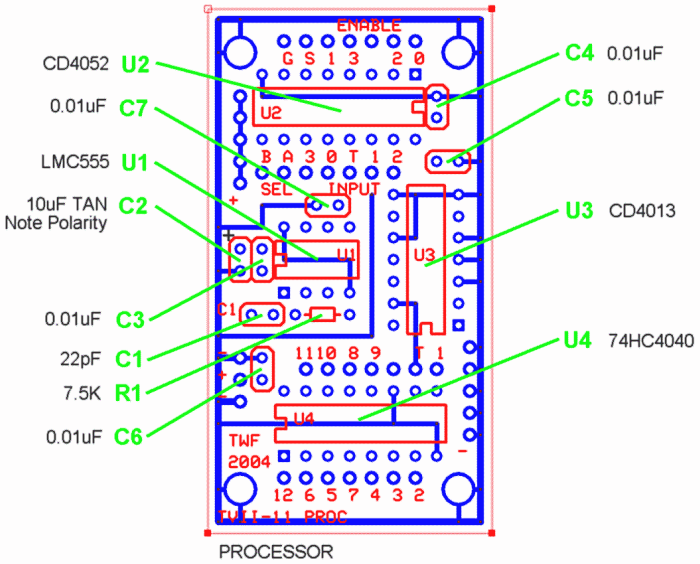



TWF Power Electronics
ThereminVision - II Sensor
Terry Fritz
Version 11
3/20/2004
Sheet 1 / 1
+5 Volts
OUTPUT
ANTENNA
GROUND
GROUND
ENABLE
C2
10uF TAN
C3
10nF
R3
100K
C1
22pF NPO
R2
10K
R1
10K
C4
10nF
U1
LMC555
+V
OUT
-TRIG
THRESH
GND
-RST
CONT
8
3
2
6
1
4
5

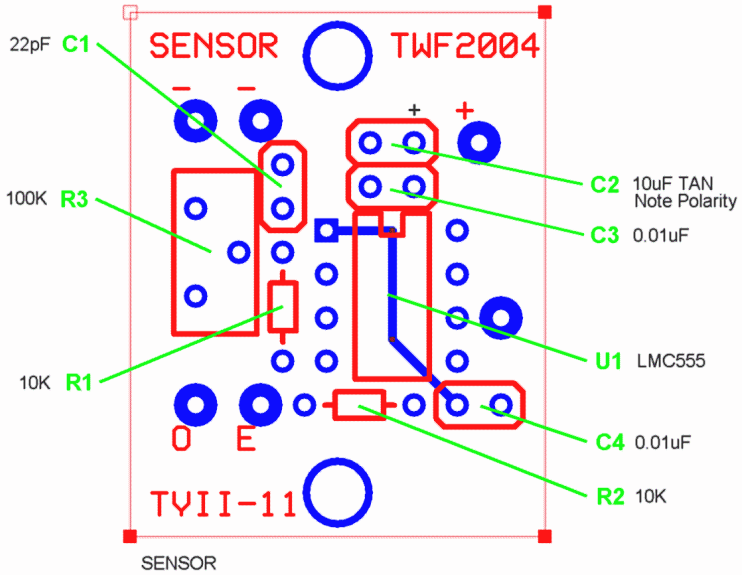
Parts List

ThereminVision - II	Designator	Description	DigiKey #	Quantity
PROCESSOR				
	C1	Cap 22pF 5% NPO 0.1 inch	P4841-ND	1
	C2	Cap 10uf 10V Tantalum 0.1 inch	P2026-ND	1
	C3 - C7	Cap 10nF 10% XTR Ceramic 0.1 inch	P4922-ND	5
	R1	Res 7.5K 5% 1/8 Watt	7.5KEBK-ND	1
	U1	LMC555CN	LMC555CN-ND	1
	U2	CD4052 CMOS	CD4052BCN-ND	1
	U3	CD4013 CMOS	CD4013BCN-ND	1
	U4	74HC4040 CMOS	MM74HC4040N-ND	1
	PC Board			1
SENSOR				
	C1	Cap 22pF 5% NPO Ceramic 0.1 inch	P4841-ND	1
	C2	Cap 10uf 10V Tantalum 0.1 inch	P2026-ND	1
	C3 - C4	Cap 10nF 10% XTR Ceramic 0.1 inch	P4922-ND	2
	R1 - R2	Res 10K 5% 1/8 Watt	10KEBK-ND	2
	R3	Res 100K 10% 1/4 inch Trimmer	490-2372-ND	1
	U1	LMC555CN	LMC555CN-ND	1
	PC Board			1

Component Placement



PROCESSOR



SENSOR

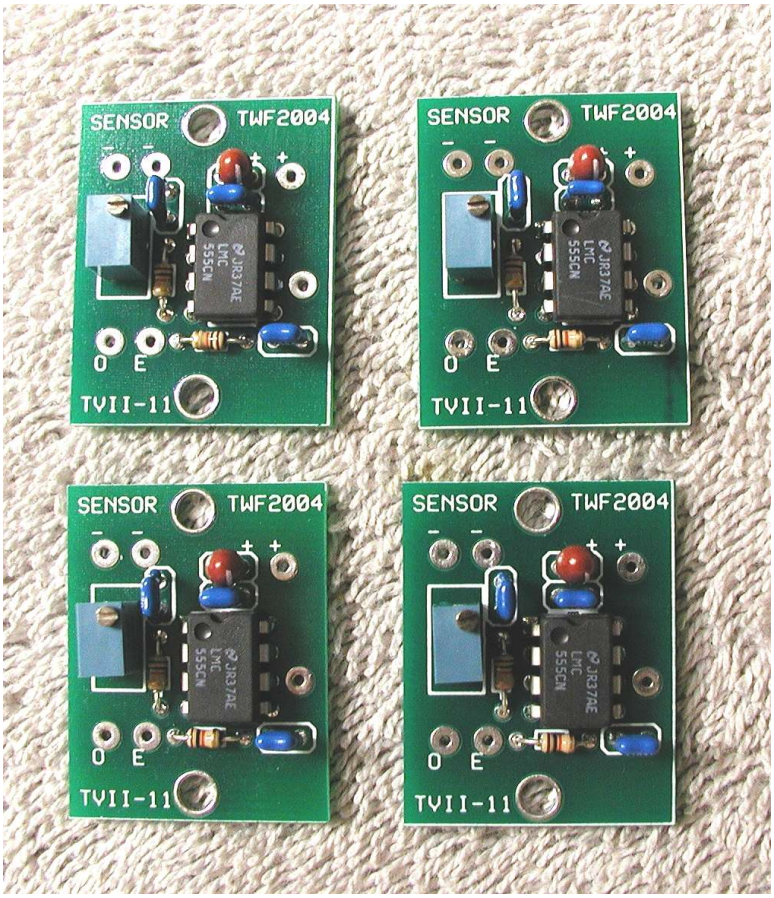
Componenet Placement Top View

Assembly Instructions

The ThereminVision-II boards can be assembled in any order. There are no critical assembly steps involved. Only typical soldering tools are needed.

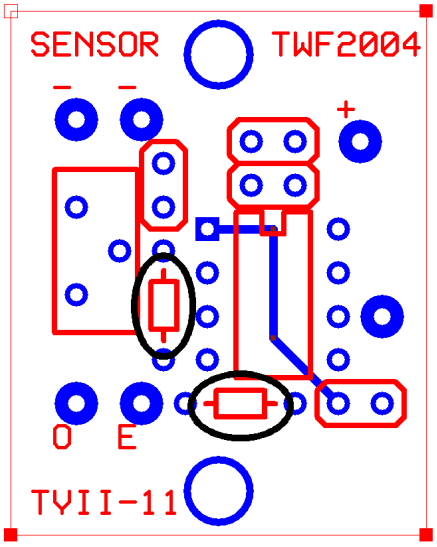
NOTE!! The processor's 7.5K timing resistor and the five 22pF timing capacitors should be separated from the other parts since they are easily mistaken for the other resistors and capacitors. The orientation of the ICs on the processor board should also be carefully noted since one is sideways and the other is upside down. All parts are inserted from the top side of the board.

Step by Step Assembly of Sensor PC Boards

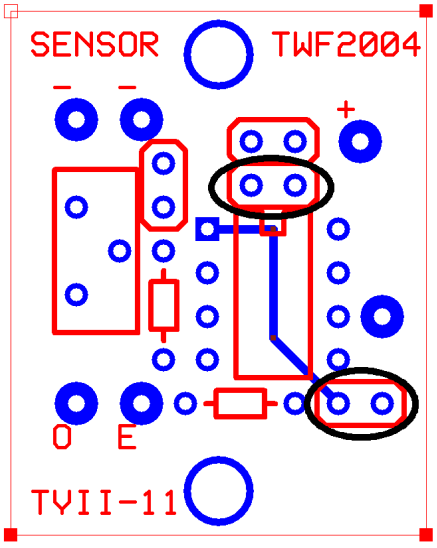


1. Locate the four smaller sensor PC boards. These will be assembled together.

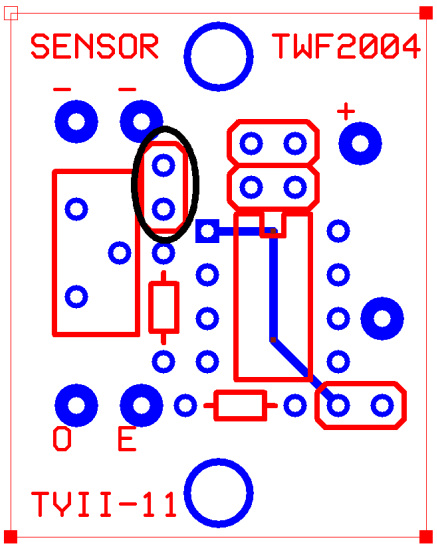
2. Install two 10K resistors (Brown-Black-Orange-Gold) in positions R1 and R2 as shown on each sensor board.



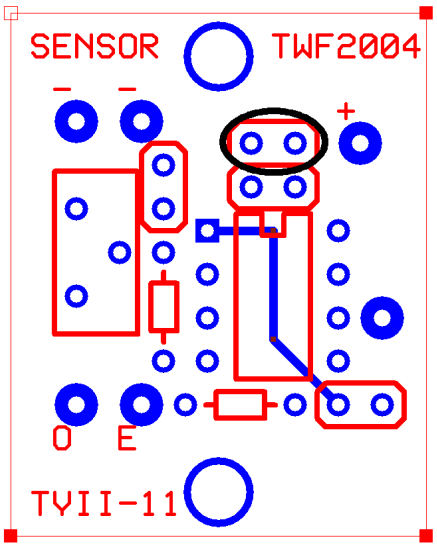
3. Install two 0.01uF (103K) capacitors in positions C3 and C4 as shown on each board.



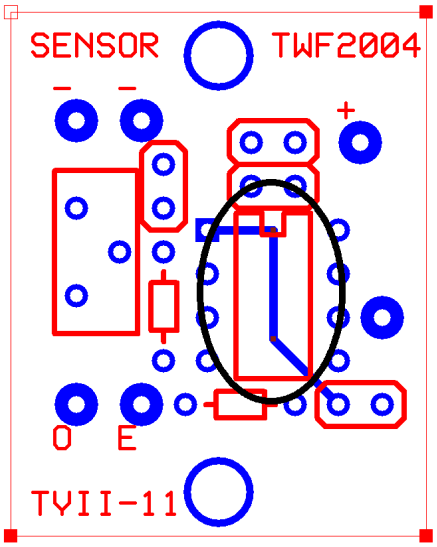
4. Install one 22pF (220J) timing capacitor in position C1 on each board.



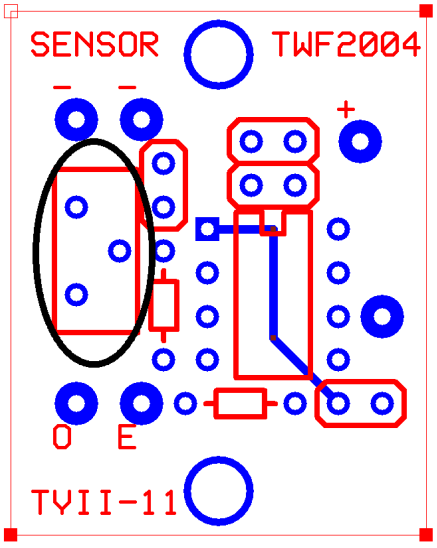
5. Install one 10uF tantalum capacitor on each board as shown. Note that the white vertical bar on the capacitor denotes the positive side of the capacitor. The positive lead with the white bar should go to the right next to the (+) on the board.



6. Install the LMC555 ICs on each board as shown. Note the proper orientation of the IC on the PC board.

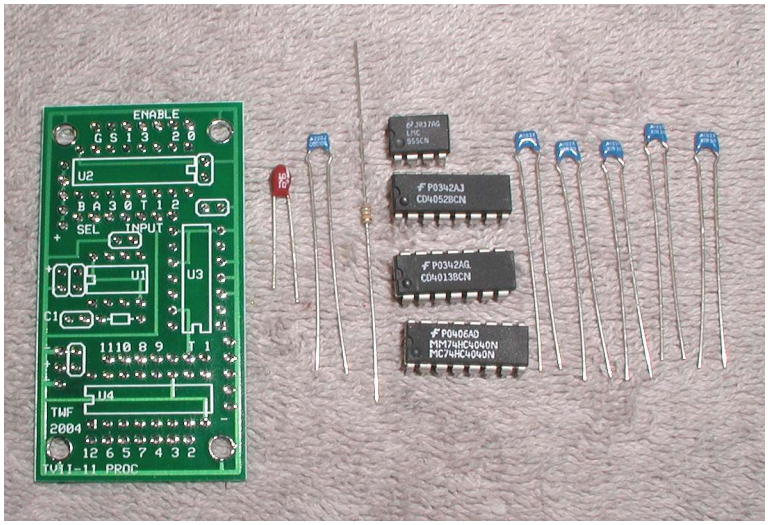


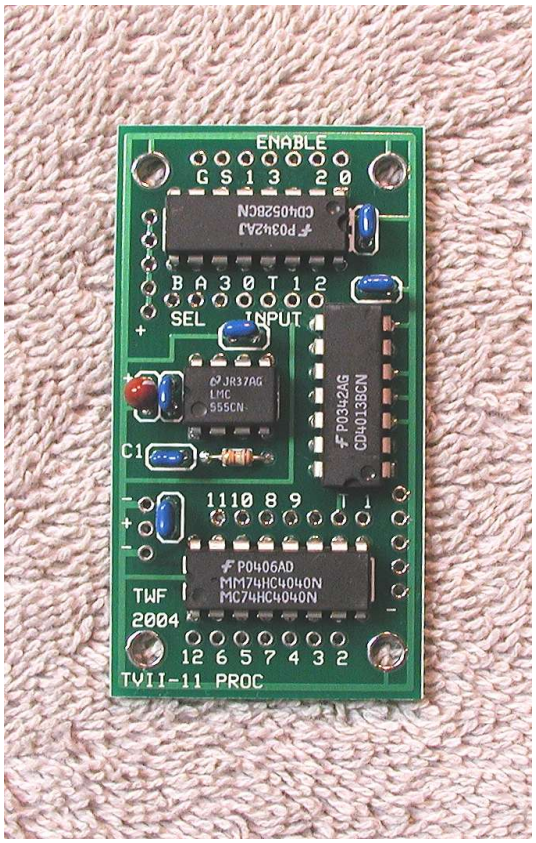
7. Install the pots in each board as shown.



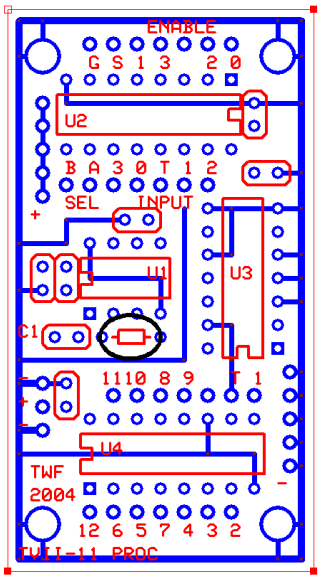
This completes the assembly of the sensors. Soldering flux should be cleaned from the boards with alcohol and a small brush.

Step by Step Assembly of Processor PC Board

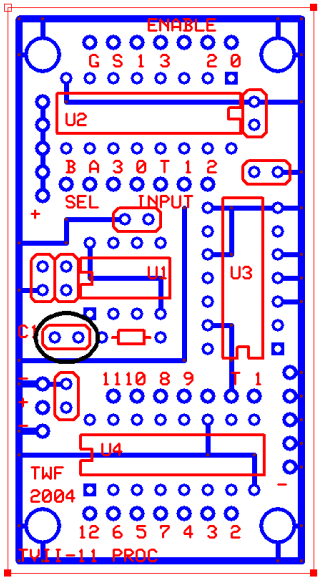




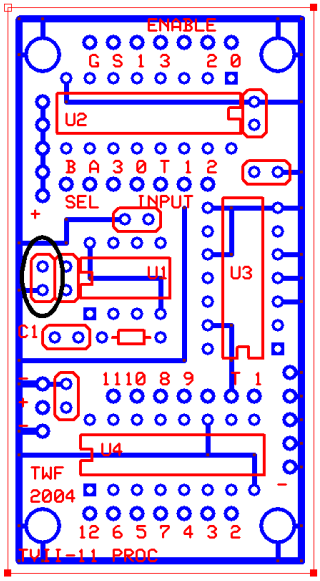
1. Install one 7.5K R1 timing resistor (Violet-Green-Red-Gold) as shown below.



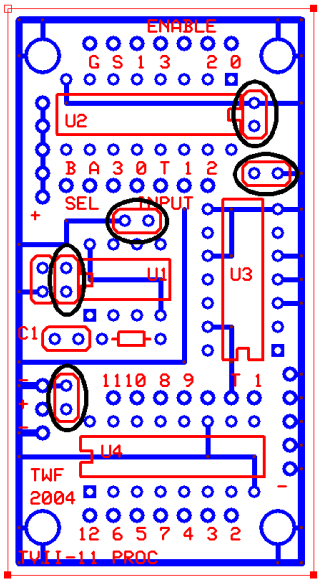
2. Install one 22pF (220J) timing capacitor C1 as shown below.



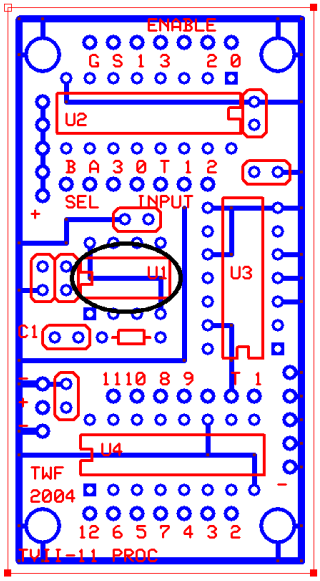
3. Install one 10uF tantalum capacitor C2 as shown. Note that the white (+) bar on the capacitor should be up to the (+) on th PC Board.



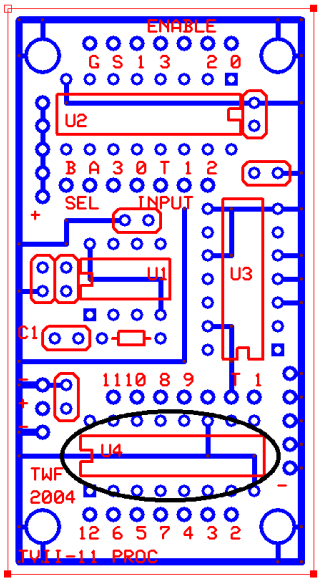
4. Install five 0.01uF (103K) capacitors as shown.



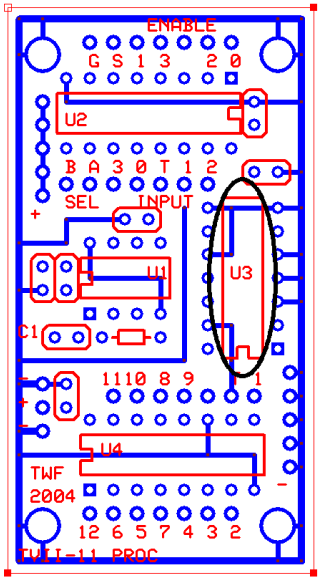
5. Install the LMC555 IC in the U1 position as shown. Note the orientation of pin one on the IC and the square pad on the board.



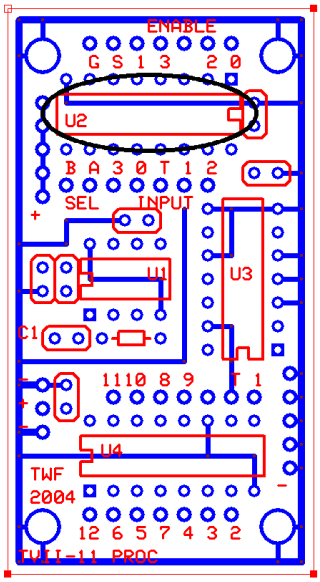
6. Install the MM74HC4040 IC in the U4 position as shown noting the orientation.



7. Install the CD4013 IC in position U3 as shown being sure to properly orient pin 1 of the IC. Note how pin one is to the left of the board.



8. Install the CD4052 IC in position U2 as shown being sure to properly orient pin 1 of the IC. Note that this IC is upside down with pin one in the upper right corner.



This completes the assembly of the processor. Soldering flux should be cleaned from the board with alcohol and a small brush.

Trouble Shooting

No pulse output from processor.

Check microcontroller programming to insure that a programming error is not causing the pulse to not be seen. This includes insuring that the pulse input pin is set for input and the three select signals to the ThereminVision-II processor are correctly set.

Check to be sure that the processor is getting +5 volts and the grounding is correct.

Check the wiring to the sensors to insure that the four wires to each sensor are properly connected.

Try setting the adjustment pots on the sensors to the center position.

Pulse width increases when an object is brought near the antenna instead of decreasing.

The sensor's oscillators are set to a higher frequency than the processor's reference oscillator. Readjust the pots on the sensor until an approaching object causes the output pulse to shorten instead of increase.

One or more sensors gives improper readings.

Check the select and signal wires to the sensors to insure they are wired correctly.

Check pot adjustments.

Check for loose parts around the antennas.

Readings drift or are unstable.

Be sure surrounding parts are solid and do not move in relation to the antennas.

Be sure soldering flux is removed form the boards with alcohol and a small brush. Flux is slightly conductive and can foul the readings.

References

The ThereminVision web site is at:
www.thereminvision.com

The ThereminVision-II website is at:
www.thereminvision.com/version-2/TV-II-index.html

There is a yahoo group for ThereminVision:
http://groups.yahoo.com/group/thereminvision/

Vendor site:
www.robotlandinc.com

Terry Fritz's E-mail
thereminvision2@thereminvision.com

Latest manual version:
http://thereminvision.com/version-2/ThereminVision-II-manual.pdf

FieldTheory robot control program as of April 15, 2004

```
'{$STAMP BS2p}
'{$PBASIC 2.5}
'Theremin Vision Robot Control Program
'Revision 200  April 11, 2004
'
'
'This program controls a Theremin Vision robot designed to compete in
'the Critter Crunch 2lb. division.


'====VARIABLE SETUP=======================================================
edgetrigger CON 15       'edge detect level - edgetrigger should be less than object
trigger
objecttrigger CON 30     'object detect level
calareatime CON 1000     'mSec to get to cal area
calchance CON 4          '*chance of calibration zeroing x/256
backtime CON 2           '*backoff time
calrep CON 10            'calibration repetitions
                         '* = loop time dependant 66 cylcles/sec.

bmid CON 32767           'two's compliment used alot
incal VAR Bit            'calibration flag
chance VAR Byte          'random chance variabe
edgedetect VAR Bit       'edge detection flag
objectdetect VAR Bit     'object detection flag
edge VAR Nib             'edge detect
object VAR Nib           'object detect
calcount VAR Nib         'count cal cycles
antenna VAR OUTD         'antenna to pin mapping

pulseget CON 9           'pulse input pin mapping per 74HCT4040 division
                         'Pin 8 = / 256
                         'Pin 9 = / 512
                         'Pin 10 = / 1024
                         'Pin 11 = / 2048
led0 PIN 6               'LED pin mapping left RED
led1 PIN 5               'right YELLOW
led2 PIN 4               'far right YELLOW

controlswitch PIN 7      'control switch pin mapping.  control switch is
                         'hard wired goto 0 state when pressed.  there is
                         'no debounce!
senseFL VAR Word         'sensor input Front Left
senseFR VAR Word         'sensor input Front Right
senseRL VAR Word         'sensor input Rear Left
senseRR VAR Word         'sensor input Rear Right
offsetFL VAR Word        'sensor offset calibrations
offsetFR VAR Word
offsetRL VAR Word
offsetRR VAR Word

motor VAR OUTA           'motor control nib  3210
                         '                   ||||
                         '                   ||||--Left Forward pin 0
                         '                   |||---Left Reverse pin 1
                         '                   ||----Right Forward pin 2
                         '                   |-----Right Reverse PIN 3
                         '
                         '
mstop CON 0              '  0  0000  stop
mleftfor CON 1           '  1  0001  left forward
mleftrev CON 2           '  2  0010  left reverse
                         '  3  0011  na
mrightfor CON 4          '  4  0100  right forward
mforward CON 5           '  5  0101  forward
mspinleft CON 6          '  6  0110  spin left
                         '  7  0111  na
mrightrev CON 8          '  8  1000  right reverse
mspinright CON 9         '  9  1001  spin right
```

```
mreverse CON 10          '  10  1010  reverse
                         '  11  1011  na
                         '  12  1100  na
                         '  13  1101  na
                         '  14  1110  na
                         '  15  1111  na


'=====SYSTEM SETUP============================================================

DIRA = 15                                'set motor drive pins for output
'DIRA=0                                  'disable motors
motor = 0                                'turn drive off now
DIRB = 7                                 'pin4 is LED0, pin5 is LED1, pin6 is LED2, and
pin7 is button
led0 = 0                                 'turn LED0 off
led1 = 0                                 'turn LED1 off
led2 = 0                                 'turn LED2 off
DIRC = 0                                 'set pulse input pins
DIRD = 7                                 'set antenna control pins
antenna = 4                              'turn antennas off
incal = 0                                'system not in calibration
objectdetect = 0                         'no objects detected
edgedetect = 0                           'no edges detected
chance = 85                              'seed random chance variable

'=====PLATFORM CONTROL FUNCTIONS=============================================
crunch:
led0 = 1 : led1 = 0 : led2 = 0           'led0 on
IF controlswitch = 1 THEN crunch         'wait for button press to begin fight
led0 = 0                                 'LED0 off and battle begins
motor = mforward                         'go to calibration area
PAUSE calareatime                        'time to get to cal area
calcount = 0                             'zero cal failure counter
GOSUB calibrate                          'calibrate sensors

  wander:                                'wonder around
  RANDOM chance                          'scramble chance varible
  IF controlswitch = 0 AND incal = 1 THEN motor = mstop : led0=1 : led1 = 1 : led2 = 1 :
PAUSE 2000 : GOTO crunch
  GOSUB scan                             'scan array
'GOSUB display                           'test outputs - normally off
  GOSUB objectfind                       'check for object
   IF objectdetect = 0 THEN GOSUB edgefind 'check for edge
   led1 = edgedetect : led2 = objectdetect
   IF edgedetect = 0 AND objectdetect = 0 AND chance < 5 THEN motor = mforward
   GOTO wander
END


'=====SUBROUTINES============================================================
calibrate:
led1 = 1 : led2 = 1                        'calibrate the sensors
incal = 0
offsetFL = 0                               'zero offsets
offsetFR = 0
offsetRL = 0
offsetRR = 0
FOR calcount = 1 TO calrep                      'scan array five times
 GOSUB scan
 offsetFL = offsetFL + senseFL             'find the offsets
 offsetFR = offsetFR + senseFR
 offsetRL = offsetRL + senseRL
 offsetRR = offsetRR + senseRR
 PAUSE 25
NEXT
 offsetFL = offsetFL/calrep                'average over many scans
 offsetFR = offsetFR/calrep
 offsetRL = offsetRL/calrep
 offsetRR = offsetRR/calrep
 incal = 1
 led1 = 0 : led2 = 0
RETURN
```

```
scan:                                   'scan the antenna array
 led0 = 1
 antenna = 0                            'turn on the sensor
 PAUSE 1
 PULSIN pulseget,1,senseRL              'measure the pulse width
'STOP                                   'stop with it on for testing
 antenna = 1
 PULSIN pulseget,1,senseFR
'STOP
 antenna = 3
 PULSIN pulseget,1,senseFL
'STOP
 antenna = 2
 PULSIN pulseget,1,senseRR
'STOP
 antenna = 4                            'turn off antennas
 led0 = 0

IF incal = 0 THEN RETURN

 'normalize readings
 'the readings will normally be 0.  an object will make the reading go
 'higher while a platform edge will make it go lower.
 senseFL = offsetFL - senseFL
 senseFR = offsetFR - senseFR
 senseRL = offsetRL - senseRL
 senseRR = offsetRR - senseRR
 'auto zero function
 'don't auto zero if edge or object is detected
 IF (edgedetect = 1) OR (objectdetect = 1) OR (chance > calchance) THEN RETURN
 IF senseFL > bmid THEN offsetFL = offsetFL - 1 ELSE offsetFL = offsetFL + 1
 IF senseFR > bmid THEN offsetFR = offsetFR - 1 ELSE offsetFR = offsetFR + 1
 IF senseRL > bmid THEN offsetRL = offsetRL - 1 ELSE offsetRL = offsetRL + 1
 IF senseRR > bmid THEN offsetRR = offsetRR - 1 ELSE offsetRR = offsetRR + 1
RETURN

objectfind:
  object= 0
  objectdetect = 1
  IF senseFL - objecttrigger < bmid THEN object = object + 1
  IF senseFR - objecttrigger < bmid THEN object = object + 2
  IF senseRL - objecttrigger < bmid THEN object = object + 4
  IF senseRR - objecttrigger < bmid THEN object = object + 8
  IF object = 0 THEN objectdetect = 0 : RETURN
  ON object - 1 GOSUB mrf,mlf,mf,msl,msl,o1,msl,msr,o1,msr,msr,mr,msl,msr,o1
RETURN

edgefind:                               'if there is an edge, back off
  edge = 0
  edgedetect = 1
  IF (senseFL + edgetrigger) > bmid THEN edge = edge + 1
  IF (senseFR + edgetrigger) > bmid THEN edge = edge + 2
  IF (senseRL + edgetrigger) > bmid THEN edge = edge + 4
  IF (senseRR + edgetrigger) > bmid THEN edge = edge + 8
  IF edge = 0 THEN edgedetect = 0 : RETURN
  ON edge - 1 GOSUB mrr,mlr,mr,mrf,mrr,e1,mrr,mlr,e1,mlr,mlr,mf,mlf,mrf,e1
RETURN

'motor contol functions
ms:
motor = mstop : RETURN
mlf:
motor = mleftfor : RETURN
mlr:
motor = mleftrev : RETURN
mrf:
motor = mrightfor : RETURN
mf:
motor = mforward : RETURN
msl:
motor = mspinleft : RETURN
```

```
mrr:
motor = mrightrev : RETURN
msr:
motor = mspinright : RETURN
mr:
motor = mreverse :RETURN
e1:
edgedetect = 0 : RETURN
o1:
objectdetect = 0 : RETURN


display:                                    'display sensor data to screen
 DEBUG CRSRXY,0,0
 DEBUG "Motor= ", BIN4 motor,CR
 DEBUG "Calibration= ",DEC1 incal,CR
 DEBUG "Edge Detect= ",DEC1 edgedetect,CR
 DEBUG "Object Detect= ",DEC1 objectdetect,CR
 DEBUG "Switch= ",DEC1 controlswitch,CR
 DEBUG "LED0= ",DEC1 led0,CR
 DEBUG "LED1= ",DEC1 led1,CR
 DEBUG "LED2= ",DEC1 led2,CR
 DEBUG "OffsetFL= ",SDEC5 offsetFL,CR
 DEBUG "OffsetFR= ",SDEC5 offsetFR,CR
 DEBUG "OffsetRL= ",SDEC5 offsetRL,CR
 DEBUG "OffsetRR= ",SDEC5 offsetRR,CR
 DEBUG CRSRXY,60,7,SDEC5 senseRR,"  "
 DEBUG CRSRXY,50,7,SDEC5 senseRL,"  "
 DEBUG CRSRXY,50,2,SDEC5 senseFL,"  "
 DEBUG CRSRXY,60,2,SDEC5 senseFR,"  "
RETURN
```