Blue Air

Manual + Documentation

Copyright and Trademark Notices

Credits

Conceptual Design: Jeffrey Stolet

Technical Design, Engineering, and Production: Andrew Lane

Manual Text & Graphics: Andrew Lane

Photographs: Andrew Lane

Contents

Introduction

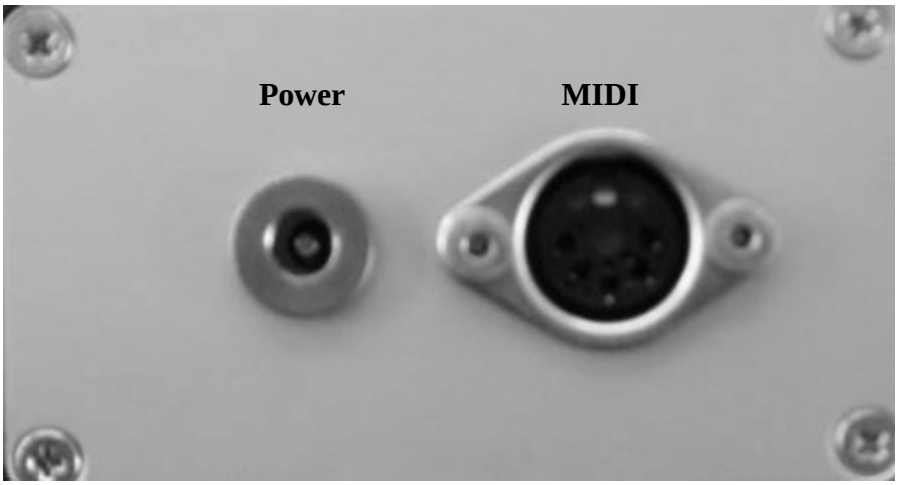
Blue Air is an infrared MIDI controller. Continuous controller information is produced movement an object or a performer's hand vertically above the infrared eye located on the top panel of Blue Air.

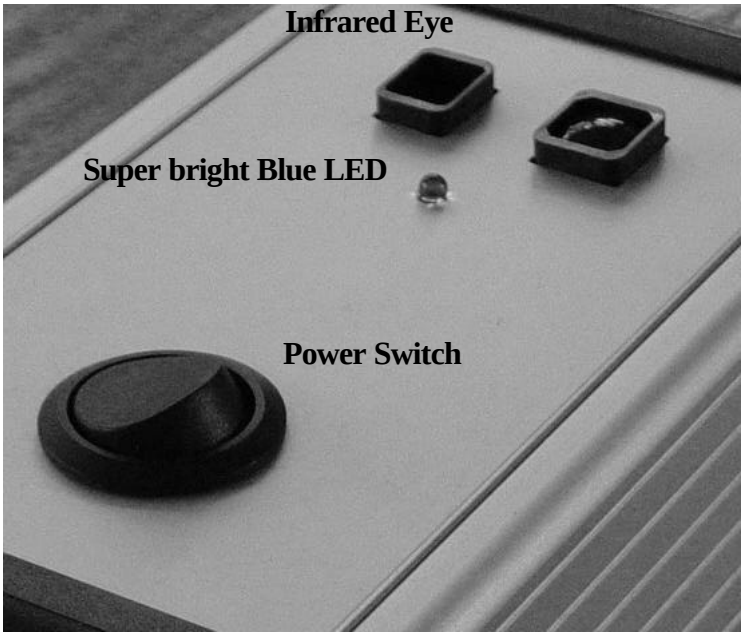
MIDI data produced by Blue Air are fast and tightly packed together. The typical time between blocks of MIDI datum is 5 milliseconds. The tremendous volume of data being sent is often best managed by MIDI manipulation software such as MAX by Cycling '74.

Blue Air is fast, accurate, stable, rugged, and simple to use.

Setup

1) Plug the power supply into an available outlet.
2) Connect the cylindrical end of the power supply to Blue Air.
3) Connect a MIDI cable to the MIDI socket (5 – din) on Blue Air. This is a MIDI OUT port.
4) Connect the other end of the MIDI cable to the desired destination for the MIDI signal (MIDI interface or instrument).
5) Turn on the power switch located on the top panel.
6) When the blue light, and the invisible infrared, beam reflect off an object (such as a hand) Blue Air will send a MIDI Controller signal out via the MIDI OUT port.





Overview

Concept and Design

Blue Air is device that outputs MIDI Continuous Controller Data derived from the distance of any object from the infrared eye located on the top panel. By placing a hand within the view of the infrared eye, a performer may create a stream of data reflecting their hand's movement within the infrared eye's view.

The circuitry is centered on Microchip's PIC16F876A microcontroller. This versatile PIC (Programmable Integrated Circuit) microprocessor contains: a programmable ROM, 4-20 Mhz speed processor, five 10-bit analog to digital converters, a USART (Universal Synchronous Asynchronous Receiver Transmitter) port, and 24 pins of digital I/O. Connected to the PIC are many small circuits that support power, MIDI, infrared sensing, and MIDI channel/controller number information.

Blue Air is contained in an anodized aluminum box, drilled and cut by a machine shop to ensure proper fit for the exterior parts. The anodized aluminum provides a smooth, polished appearance to Blue Air.

Algorithm:

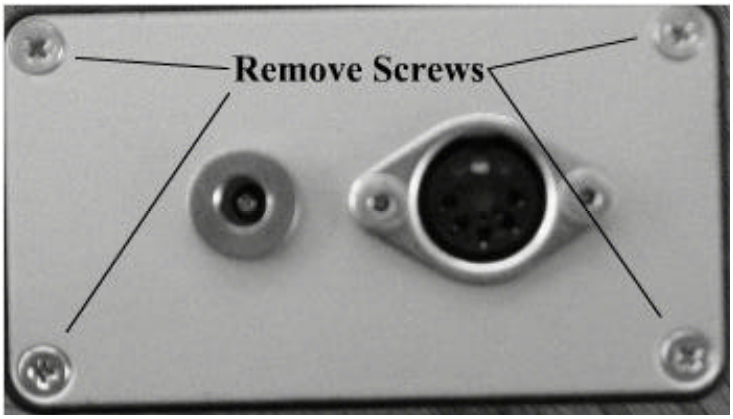
1) Blue Air is powered on and the blue light is lit.
2) The infrared eye emits an infrared beam of light in a short burst out of the recessed lens located on the top panel.
3) The infrared eye waits for the emitted infrared bandwidth of light to be detected by the elevated sensor lens. An object such as a performer's hand must be positioned over the infrared eye in order for the light burst to be detected.
4) The infrared eye records the time delay between emission and detection and expresses this time in voltage. The shorter the delay, the less the distance from infrared eye to object, and the higher the produced voltage. The longer the delay, the greater the distance from the infrared eye to the object, and the lower the produced voltage.
5) The voltage produced by the infrared eye is measured by the PIC16F876A's analog to digital converter bank and translated into a 10-bit number.
6) The programmed PIC16F876A scales the 10-bit number to a range of 0-127 and formats a digital MIDI packet based of the internal channel/controller number switch.
7) The MIDI packet leaves the PIC16F876A via the USART port and passes through a standard MIDI OUT circuit, converting the MIDI packet into a MIDI signal.
8) The MIDI signal is sent to the MIDI OUT port located on the side of Blue Air.

Special Settings

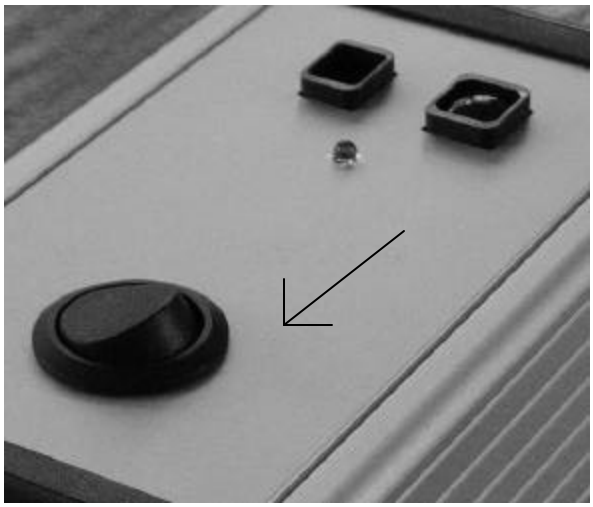
Blue Air generates a MIDI signal that is formatted as continuous controller data. The continuous controller number and MIDI channel may be set if the MIDI setup for performance demands specific controllers and MIDI channels to be utilized.

To set either the continuous controller number or MIDI channel Blue Air must be opened up to gain access to the 8-bit binary switch inside the box.

1. TURN BLUE AIR OFF!!! Remove the four screws on the connector side of the box. Loosen the side panel from the box.



2. Slide the top panel out the open end of the box. On some boxes the top screws on the opposite side will need to be loosened



3. Locate the 8-bit binary channel/controller number switch on the circuit board.

Setting the Controller Number

On the 8-bit binary switch, switches 1-4 are used to set the continuous controller number for the MIDI data. Blue Air uses LEFT-JUSTIFIED binary, so please use the following chart to set the Controller Number:

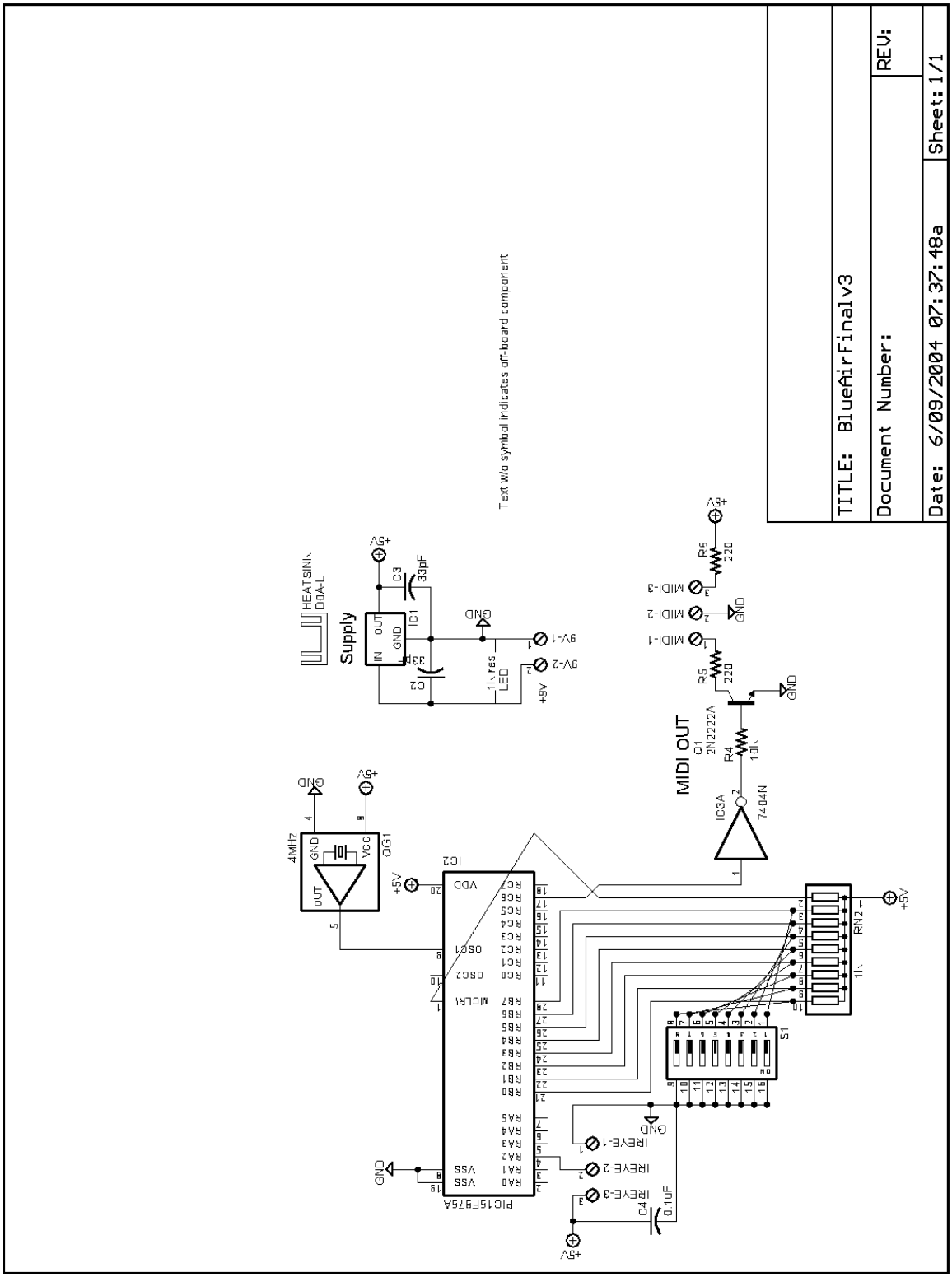
Controller Number	Switch Settings
#	1234
0	0000
1	1000
2	0100
3	1100
4	0010
5	1010
6	0110
7	1110
8	0001
9	1001
10	0101
11	1101
12	0011
13	1011
14	0111
15	1111

Setting the MIDI Channel

On the 8-bit binary switch, switches 5-8 are used to set the MIDI channel for the MIDI data. Blue Air uses LEFT-JUSTIFIED binary, so please use the following chart to set the Controller Number:

MIDI Channel	Switch Settings
#	5678
1	0000
2	1000
3	0100
4	1100
5	0010
6	1010
7	0110
8	1110
9	0001
10	1001
11	0101
12	1101
13	0011
14	1011
15	0111
16	1111

Schematic


TITLE: BlueAirFinalv3
Document Number:
REV:
Date: 6/09/2004 07:37:48a
Sheet: 1/1
Text w/o symbol indicates off-board component
HEATSINK
DOA-L
Supply
IC1
C3
33pF
C2
33pF
9V-1
9V-2
+9v
1K res
LED
MIDI OUT
Q1
2N2222A
R4
10K
R5
220
MIDI-1
MIDI-2
MIDI-3
IC3A
7404N
4MHz
OUT
GND
Vcc
OG1
IC2
PIC16F876A
MCLR
OSC1
OSC2
VDD
VSS
RN2
10K
S1
IREYE-1
IREYE-2
IREYE-3
C4
0.1UF
+5V
GND

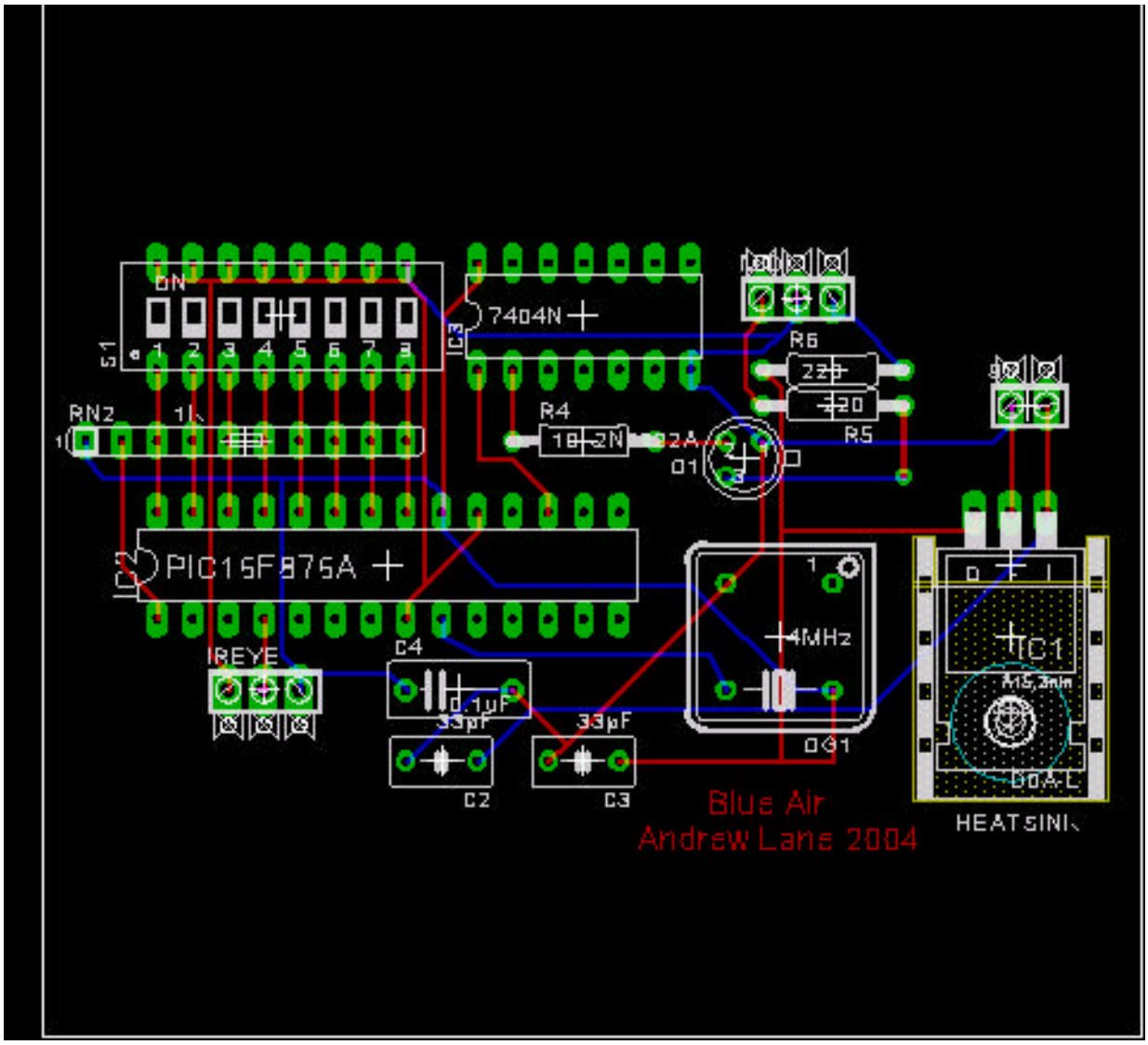
PIC Code

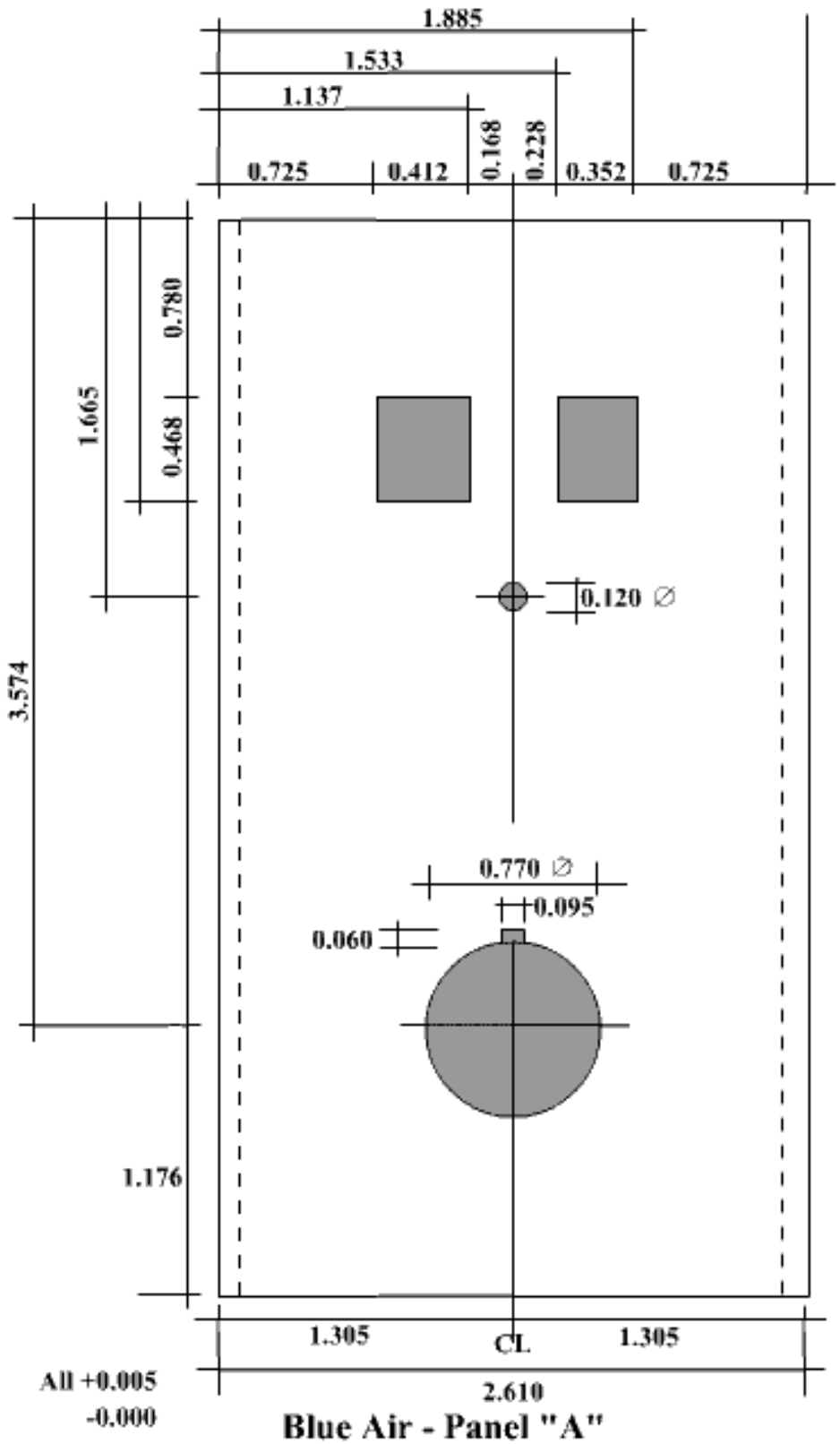
```
' -----[ Title ]-----------------------------------------------------------
'
' File..... BlueAir.BAS
' Purpose... IR sensor -> 16F876A -> MIDI
' Author.... Andrew Lane
' Started... August 14, 2003
' Updated... February 2, 2004

' -----[ Program Description ]---------------------------------------------
'
' This program uses the 16F876A to read a Sharp GP2D12 or GP2Y0A02YK IR sensor
with variable voltage output.
' The read voltage is converted to a MIDI message and sent via MIDI connection.
'
'       Pin#    PIC         Connection
'       ----    ---         ----------
'       1       MCLR        5V (thru 1K)
'       2       RA0         NC
'       3       RA1         NC
'       4       RA2         IR sensor connected (yellow)
'       5       RA3         NC
'       6       RA4         NC
'       7       RA5         NC
'       8       VSS         GND
'       9       OSC1        Clock
'       10      OSC2        NC
'       11      RC0         NC
'       12      RC1         NC
'       13      RC2         NC
'       14      RC3         NC
'       15      RC4         NC
'       16      RC5         NC
'       17      RC6         Pin 1 7404 (midi out circuit)
'       18      RC7         NC
'       19      VSS         GND
'       20      VDD         5V
'       21      RB0         Binary Switch 1
'       22      RB1         Binary Switch 2
'       23      RB2         Binary Switch 3
'       24      RB3         Binary Switch 4
'       25      RB4         Binary Switch 5
'       26      RB5         Binary Switch 6
'       27      RB6         Binary Switch 7
'       28      RB7         Binary Switch 8
```

```
'
'       PIC16F876A
'       ------\/-------
'       | 1       28 |
'       | 2       27 |
'       | 3       26 |
'       | 4       25 |
'       | 5       24 |
'       | 6       23 |
'       | 7       22 |
'       | 8       21 |
'       | 9       20 |
'       | 10      19 |
'       | 11      18 |
'       | 12      17 |
'       | 13      16 |
'       | 14      15 |
'       --------------


' -----[ Revision History ]------------------------------------------------
'
'


' -----[ Constants/Defines ]-----------------------------------------------------
'


' Define ADCIN parameters
DEFINE      ADC_BITS        10      ' Set number of bits in result
DEFINE      ADC_CLOCK       3       ' Set clock source (3=rc)
DEFINE      ADC_SAMPLEUS    50      ' Set sampling time in uS
DEFINE      HSER_RCSTA      90h     ' enable the receive register
DEFINE      HSER_TXSTA      20h     ' enable the transmit register
DEFINE      HSER_BAUD       31250   ' set the baud rate



' -----[ Variables ]--------------------------------------------------
'
adval       VAR WORD                ' Create adval to store result
midiout     VAR BYTE                ' Create midiout value variable
midioutOLD  VAR BYTE                ' Create midiout value variable
chnum       VAR BYTE                ' Create channel number variable
ctrlnum     VAR BYTE                ' Create controller number variable
```

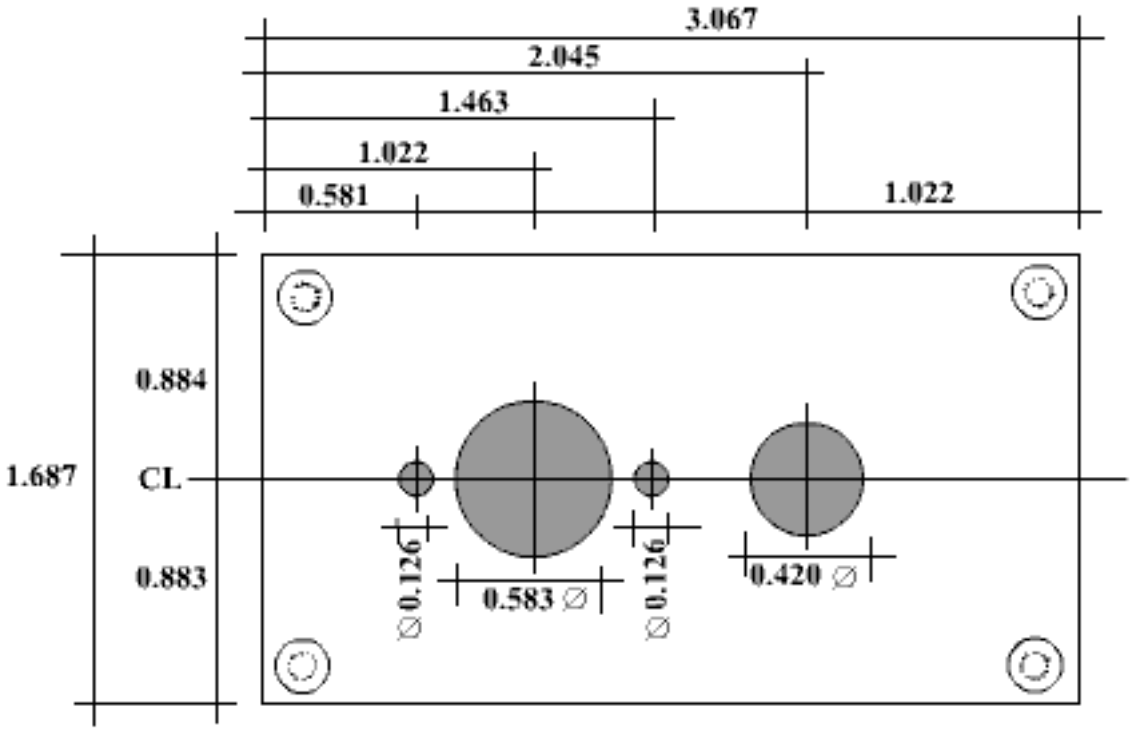
```
' -----[ Initialization ]----------------------------------------------------
'
Init:
        PORTB = $00             ' all outputs off to start
        TRISB = %11111111       ' Set PORTB to digital input

        TRISA = %00000100       ' Set PORTA.2 to input
        ADCON1 = %10000010      ' Set PORTA analog - right justified


' -----[ Main Code ]-----------------------------------------------------
'




' -----[ Read Channel Number ]-------------------------------------------
'

        chnum = %10110000

        IF PORTB.7 = 0 Then
                chnum = chnum + 8
                EndIF

        IF PORTB.6 = 0 Then
                chnum = chnum + 4
                EndIF

        IF PORTB.5 = 0 Then
                chnum = chnum + 2
                EndIF

        IF PORTB.4 = 0 Then
                chnum = chnum + 1
                EndIF


' -----[ Read Controller Number ]----------------------------------------
'

        ctrlnum = %00000000

        IF PORTB.3 = 0 Then
                ctrlnum = ctrlnum + 8
                EndIF
```

```
        IF PORTB.2 = 0 Then
                ctrlnum = ctrlnum + 4
                EndIF

        IF PORTB.1 = 0 Then
                ctrlnum = ctrlnum + 2
                EndIF

        IF PORTB.0 = 0 Then
                ctrlnum = ctrlnum + 1
                EndIF


' -----[ Loop ]------------------------------------------------------------
'

loop:

' -----[ Read IR Sensor ]--------------------------------------------------
'

        ADCIN 2, adval                  ' Read RA2 to adval

        adval = adval / 3
        adval = adval - 40
        midiout = adval


' -----[ Send MIDI ]-------------------------------------------------------
'

        IF midiout == midioutOLD Then
                GoTo same
        EndIF
        HSerout [chnum, ctrlnum, midiout]
        midioutOLD = midiout

' -----[ same ]------------------------------------------------------------
'

same:
        GoTo loop

' -----[ End Loop ]--------------------------------------------------------
'
        End
```


DN
S1
1 2 3 4 5 6 7 8
7404N
IC3
RN2
1k
R4
18
2N
2A
Q1
R6
220
R5
PIC16F876A
IC2
REYE
C4
0.1µF
33pF
C2
33pF
C3
4MHz
Q1
IC1
HEATSINK
Blue Air
Andrew Lane 2004

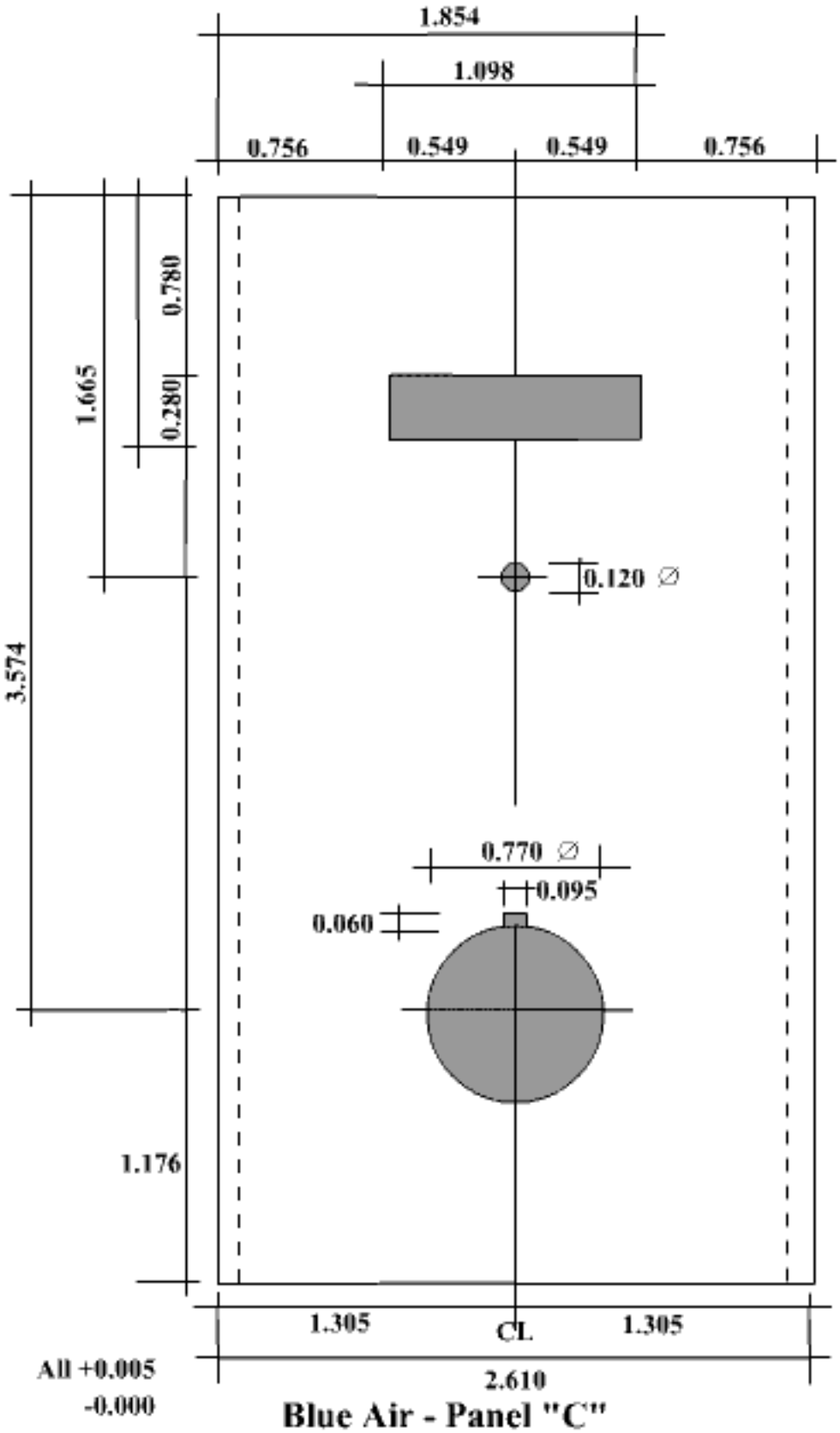


Blue Air - Panel "A"



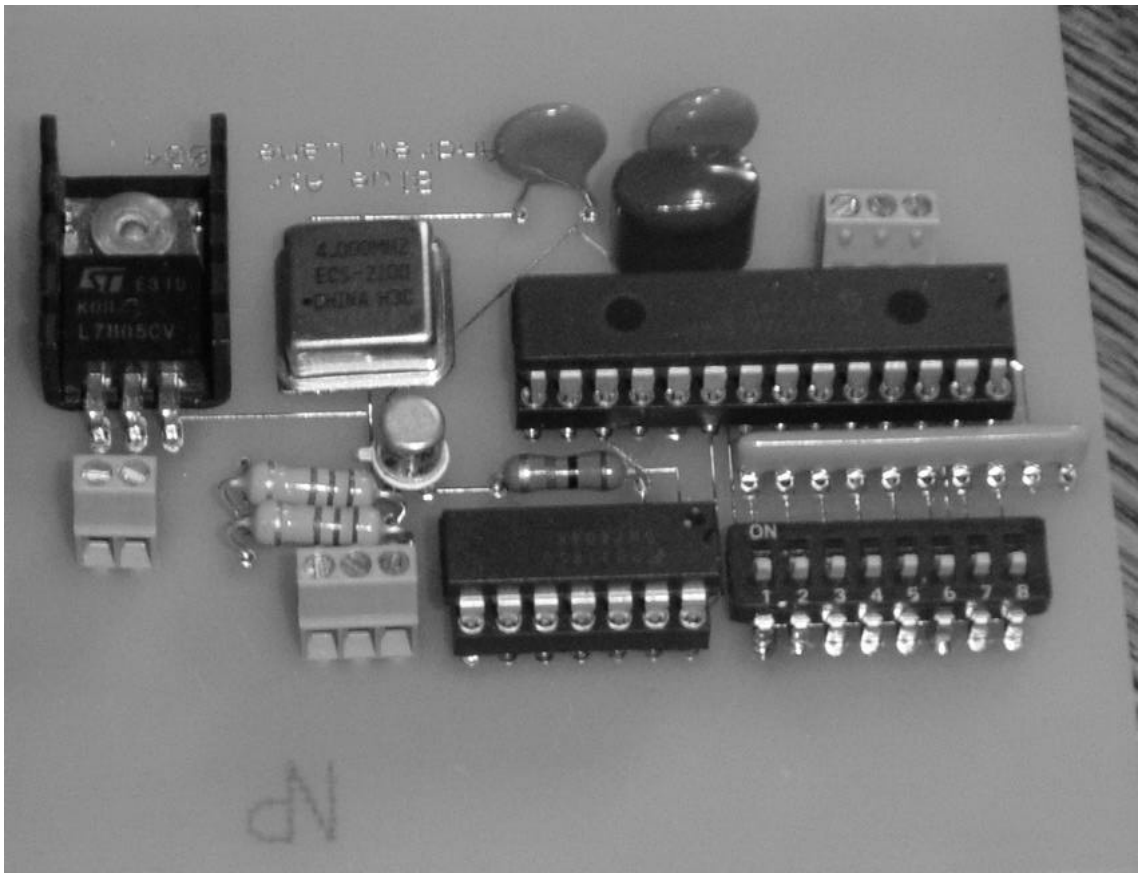
All +0.005
-0.000

Blue Air - Panel "B"

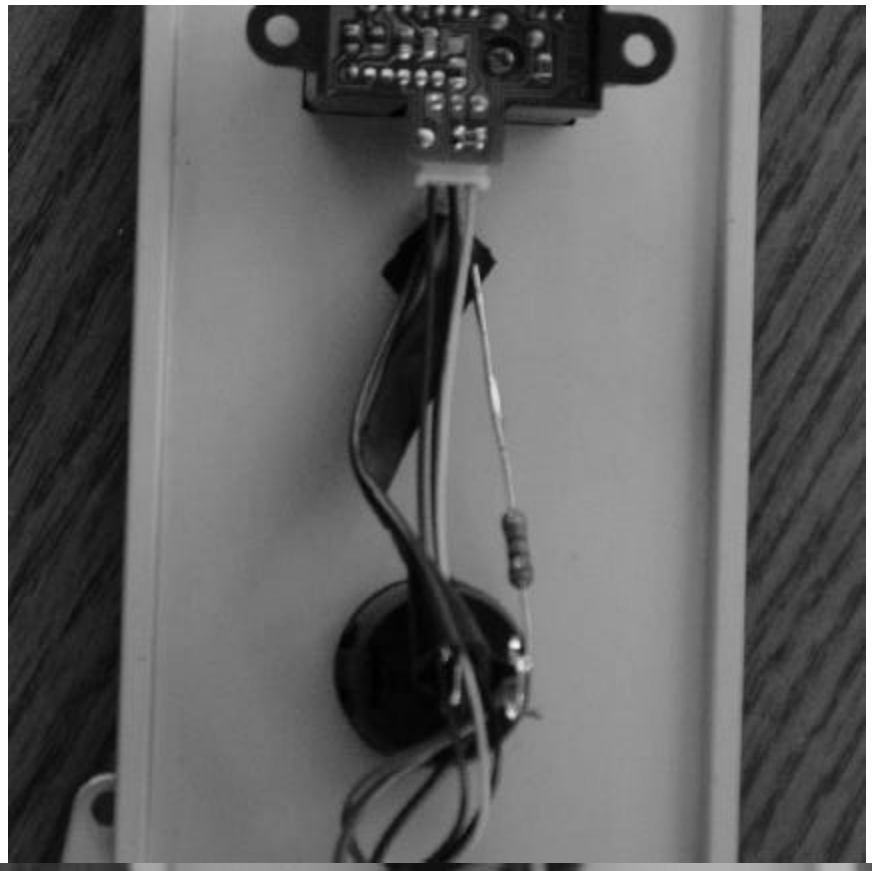


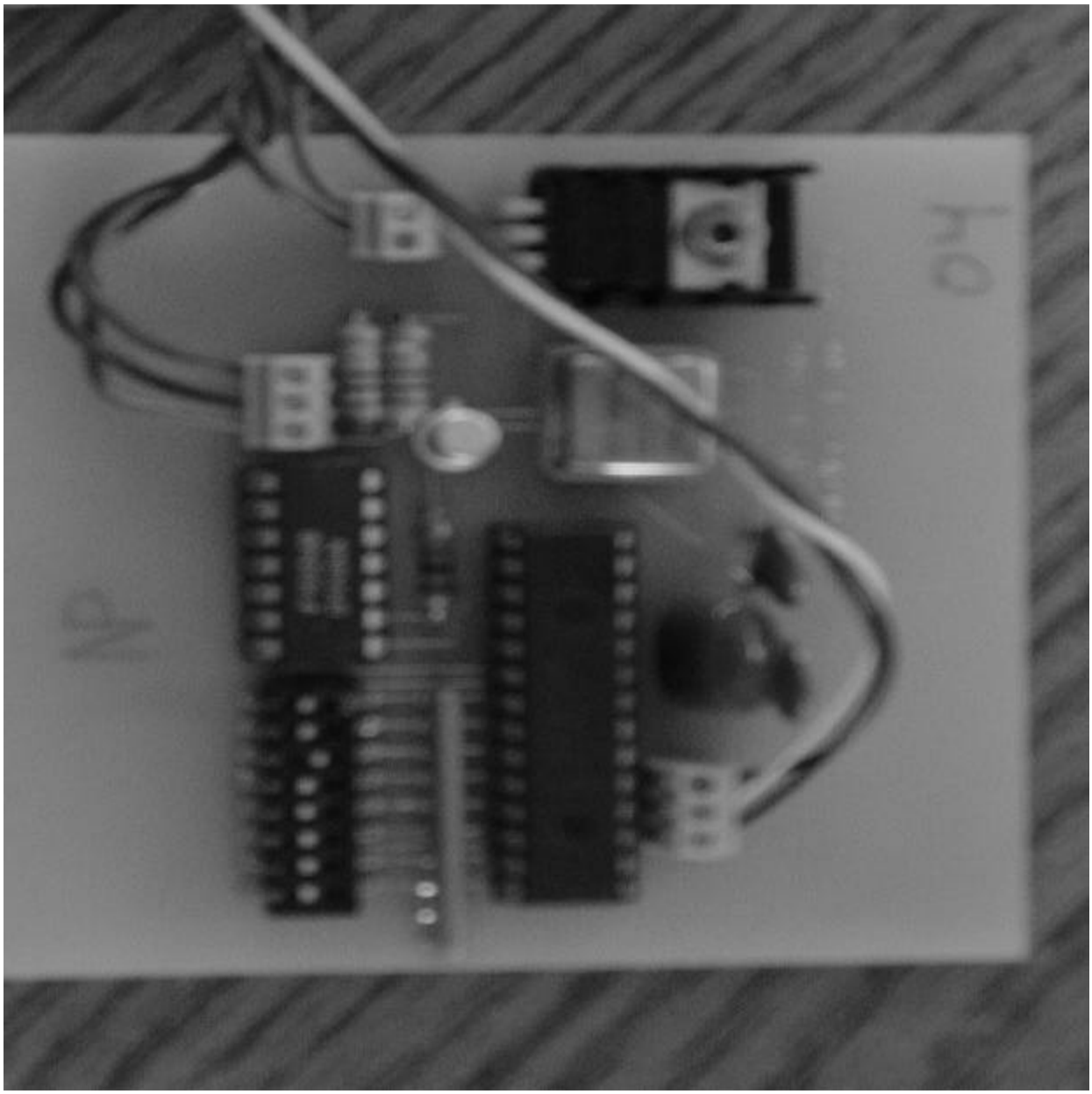
Blue Air - Panel "C"

Soldered Circuit Board



Wired Blue Air





Parts List

1	PIC16F876A
1	5-din socket
1	1K ResNetwork
1	10K Res
1	1K Res
2	220 Res
1	8 pin p-dip switch
1	7404 hex inverter
1	4 Mhz powered 1/2 size osc.
1	7805 voltage regulator
1	heatsink for 7805
1	2N2222A transistor
1	22mF Cap
2	33pF Cap
1	14 pin IC socket
1	28 pin IC socket
1	3 mm Blue LED
1	Rocker Switch
1	DC power jack 2.1 x 5.5 mm
1	9v 300 mA DC power supply
1	Hammond 1455K1201 enclosure

Technical Specifications

Name: Blue Air
Design: By Andrew Lane and Jeffrey Stolet at University of Oregon, 2003—2004.

Size: 4.72” x 3.07” x 1.69”
Power: 9vDC, 500 mA, positive pin, ground sleeve

Output: MIDI
Data Type: MIDI Continuous Controller Data
Update time: 5 ms.
Senor range: 150cm

Notes