



Audience Opinion Meter

Introduction

ADB I/O can be used to get user input from remotely placed keypads to the Mac. This can be used in behavioural experimentation, audience opinion measurements and elections in classroom setups but also in other applications like entry control of rooms in combination with electric door locks.

Example: 20 people sitting in a room could each have a keypad with three buttons. Each button would return a value that would let these 20 people "vote" on how they feel about something that they are watching (e.g. a TV show, a lawyer, a commercial, etc.).

Implementation

ADB I/O has 4 analog inputs that can be used in this application. The method used is to have the keypads generate an analog voltage that correspond to the combination of buttons pressed. Each keypad can have up to 5 buttons. Using 4 ADB I/O units up to 16 keypads could be sampled by one Mac.

By adding some circuitry to each ADB I/O, up to 32 keypads could be attached to one Mac. The circuit in *figure 1* will allow eight keypads to be attached to one ADB I/O. Since 4 ADB I/Os can be attached to one Mac, 4 similar circuits will allow a total of 32 keypads to be used.

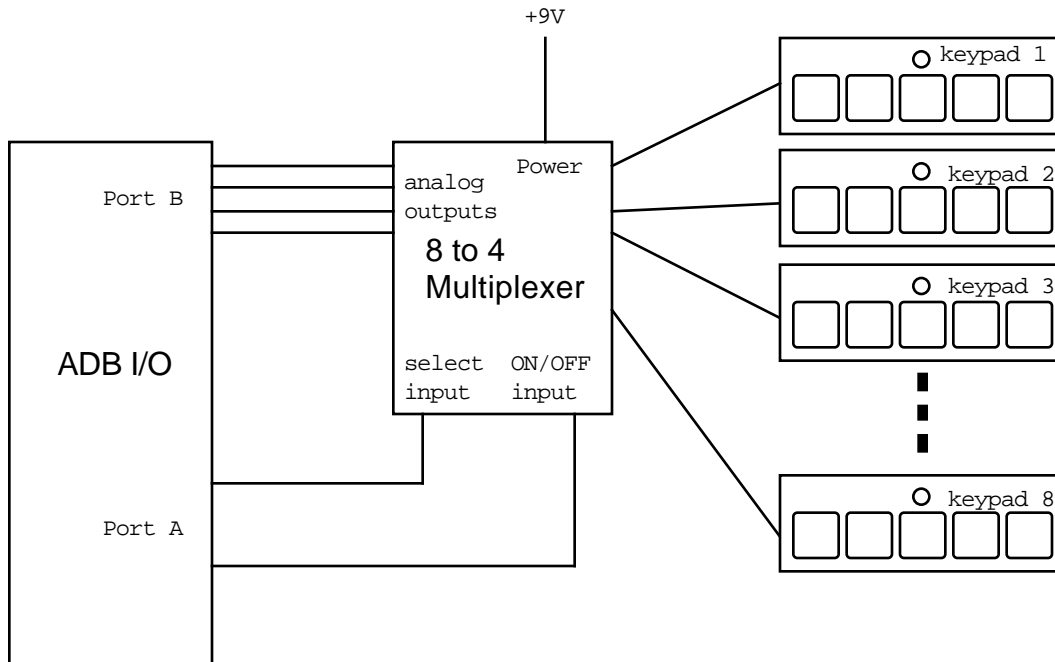


FIGURE 1. Block Diagram

The Keypad

The keypad is a box with 5 pushbuttons. A LED indicates when voting is in progress. A 3.5 mm stereo phono receptacle is used to connect the cable to the Mac.



PHOTO 1. Keypad box.

Inside the keypad is the circuit in *figure 2*. Across pin 1 and 3 of the connector J1 a voltage of +9V is applied. The LED D1 will turn on indicating that a vote is in progress and that the user should press a button.

Pressing one or several of the buttons SW1 to SW5, a current dependent of the combination of resistors R2 to R6 is fed to pin 2 of J1. This current will be translated to a voltage on the ADB I/O across the pull-down resistors on Port B.

By using shielded twisted pair cable between the keypad and the ADB I/O the design will have high noise immunity and cable length could reach 20 meters. The twisted pair should connect to pin 1 and 2 of J1 and the shield to pin 3.

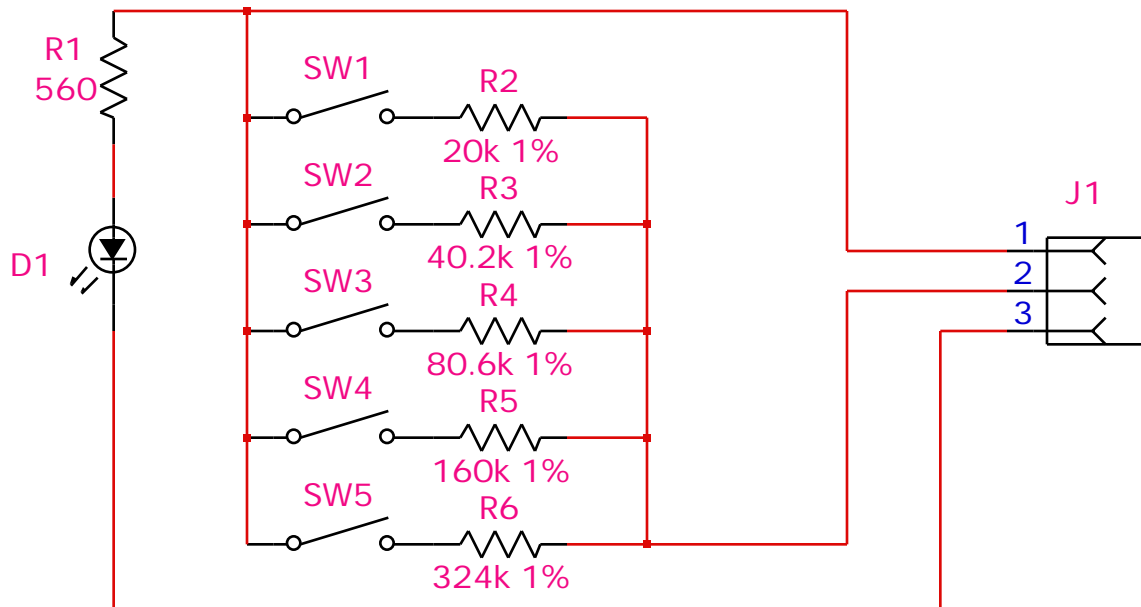


FIGURE 2. Keypad circuit.

The keypad can be easily assembled with soldered wires without any special circuit boards.

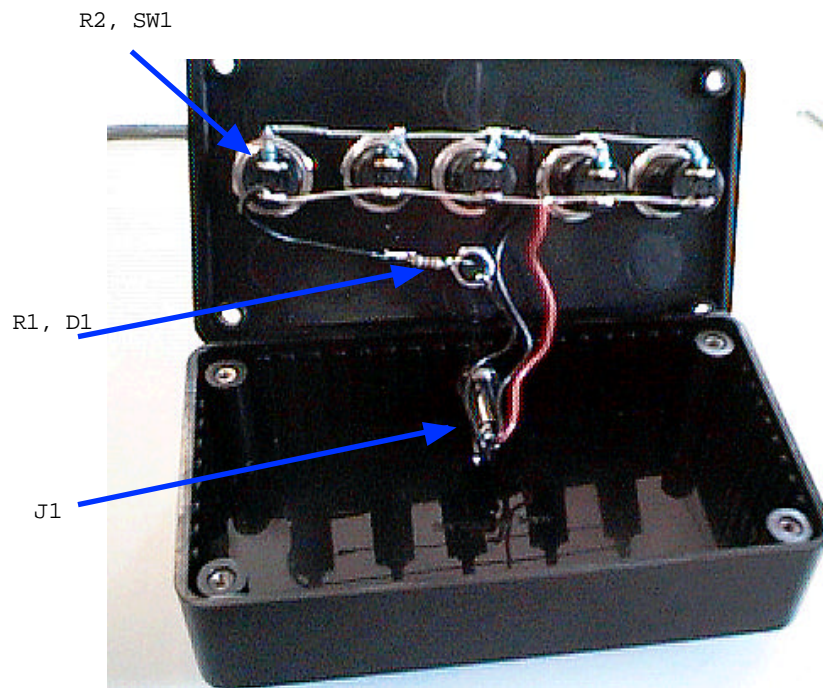


PHOTO 2. Inside Keypad

Multiplexer

At the Mac the circuit in *figure 3* will handle the multiplexing of 8 keypads to 4 analog inputs. To handle up to 32 keypads, 4 circuits can be used.

The multiplexer consists of one relay with 4 changeover contacts. A fly back diode across the relay coil will protect the ADB I/O relays from voltage spikes. A 9V DC adapter will provide voltage for the relay and the keypads. The Pull-Up/Pull-Down-Jumpers of ADB I/O port B should be be positioned for Pull-down for all four channels.

Before a vote is started ADB I/O Port B should be configured as 4 analog inputs. To start a vote channel 1 of port A is set high. Thus +9V will be fed to the keypads and the LEDs will turn on, indicating that the participants should press a button (or combinations of buttons).

Now, the ADB I/O port B is read and the values are saved.

Next, channel 2 of port A is set high. The relay will switch in the next group of four keypads and the ADB I/O port B is read again and the values are saved.

This procedure is then repeated for the other 3 circuits.

The sample rate of 32 keypads will thus be $(11\text{ms} * 2 + 11\text{ms}) * 4 = 132\text{ms}$. This assumes that the sampling is done in a tight loop which will disable the use of the mouse and keyboard during the sampling.

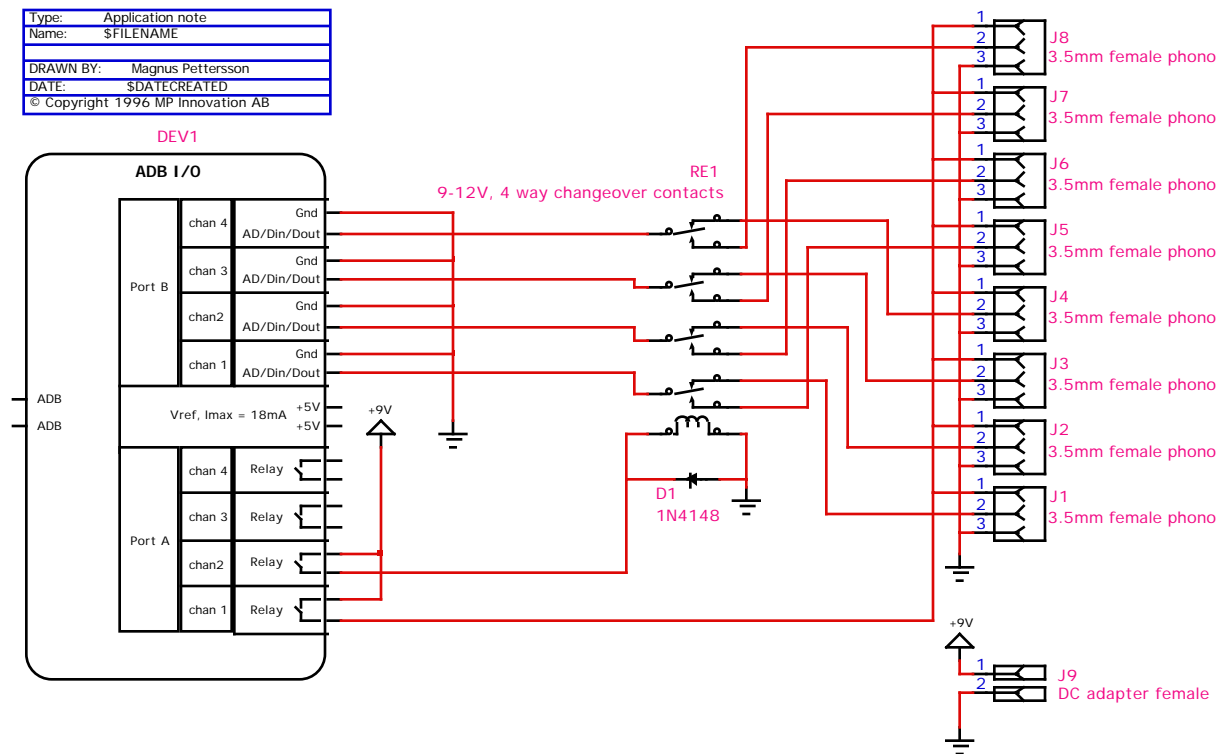


FIGURE 3. Multiplexer schematic.

Software

The values read from the keypad would ideally have a linear characteristic. But due to the 10k measurement resistor on Port B the actual values will be lower and not linear. Linearity could be achieved by using an operational amplifier to receive the current from the keypads. That is not necessary though, since a linearisation algorithm can be implemented in software.

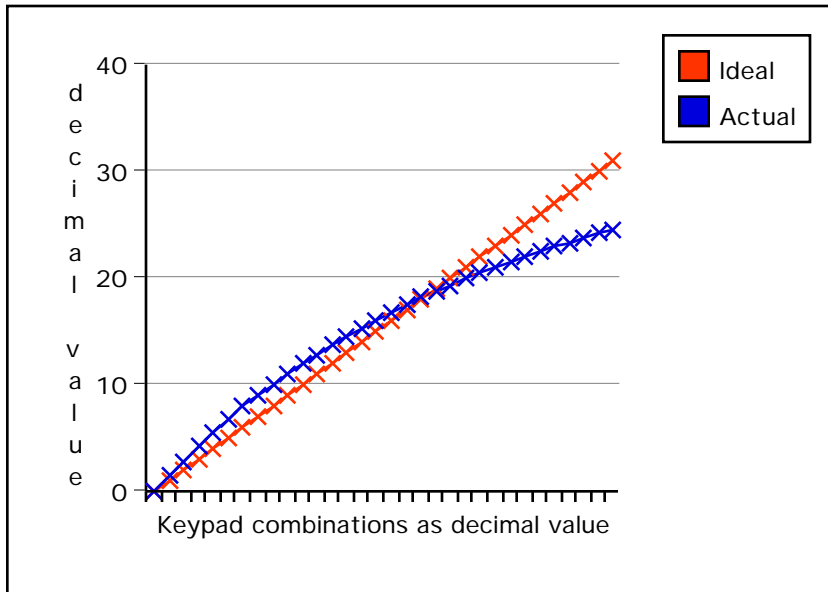


FIGURE 4. Transfer function.

By using the values in *table 1* in a conversion table, values representing all the button combinations 0 to 31 can be obtained. Using +9V as supply voltage to the keypads the measured values will range between 0 and 222. By traversing the table and finding a value * 9 that equals (with some margin) the analog value, the table index will give the linearised value.

This conversion algorithm can certainly be improved with binary search, table look up or by using a linearising formula etc., however not covered in this application note.

0
1,4344262295082
2,86885245901639
4,20081967213115
5,5327868852459
6,76229508196721
7,88934426229508
9,01639344262295
10,0409836065574
10,9631147540984
11,8852459016393
12,8073770491803
13,7295081967213
14,5491803278689
15,266393442623
15,983606557377
16,8032786885246
17,5204918032787
18,1352459016393
18,75
19,3647540983607
19,9795081967213
20,4918032786885
21,0040983606557
21,516393442623
22,0286885245902
22,4385245901639
22,9508196721311
23,3606557377049
23,7704918032787
24,1803278688525
24,5901639344262

TABLE 1. conversion table.

The code fragments in *listing 1* coded in HyperTalk illustrates how an experiment can be executed. The result will be a field of 32 values, one for each keypad. The values will be between 1 and 31 representing the combination of buttons pressed on the keypad. Single button presses will have the following values. Button 1 will have value 1, Button 2 will have value 2, Button 3 will have value 4, Button 4 will have value 8, Button 5 will have value 16.

```

on mouseUp
  put 9.0 into supply -- The supply voltage used
  repeat for 20 times -- experiment duration
    put 1 into index -- initialise result index
    configureADBIO 1, B, "analog in", "analog in", "analog in", "analog in"
    setadbio 1, A, 1 -- start experiment by turning on power

    repeat with unit = 1 to 4 -- do four ADB I/Os
      get getADBIO(unit , B) -- get value
      repeat with channel = 1 to 4 -- for all channels on port B
        get linearise(item 2 of line channel of it, supply) -- linearise values
        -- if pressed save key combination no (1..31) in the result field
        if it > 0 then put it into line index of cd field result
        add 1 to index -- next line
      end repeat
      setadbio 1, A, 2 -- switch to next 4 keypads
      get getADBIO(unit , B)
      repeat with channel = 1 to 4
        get linearise(item 2 of line channel of it, supply)
        if it > 0 then put it into line index of cd field result
        add 1 to index
      end repeat
    end repeat
  end repeat
  setadbio 1, A, 0 -- end experiment by turning off power
end mouseUp

-- This function looks up values in the table in cd fld 1
-- Values within +/- 3 is accepted.

function linearise val, voltage
  repeat with i = 0 to 31
    get voltage * the value of line (i+1) of cd fld 1
    if abs(val - it) < 4 then return i
  end repeat
end linearise

```

LISTING 1. HyperTalk code fragment.

References

1. Danielsson - Bengtsson. "Digital Teknik" 1989.
2. Hemert. "Elektronik, Digitalteknik" 1981

Acknowledgements

Thanks to Lloyd K. Komatsu and Grant Spofford for original idea and interest in this application.