Keypad Lock

Design Brief

Design and make a programmable door entry system that does not require the use of a key

Circuit Explanation

The four rows of the keypad are connected to controller output lines, and the three columns are connected to controller input lines. When no switches are pressed, there is no electrical connection between any row or column. Therefore all the inputs are at the logic level 0 (0V) status, as they are .pulled low. by the 10k resistors connected to these lines. This means the input signal to the controller is 0-0-0 along the three input lines.

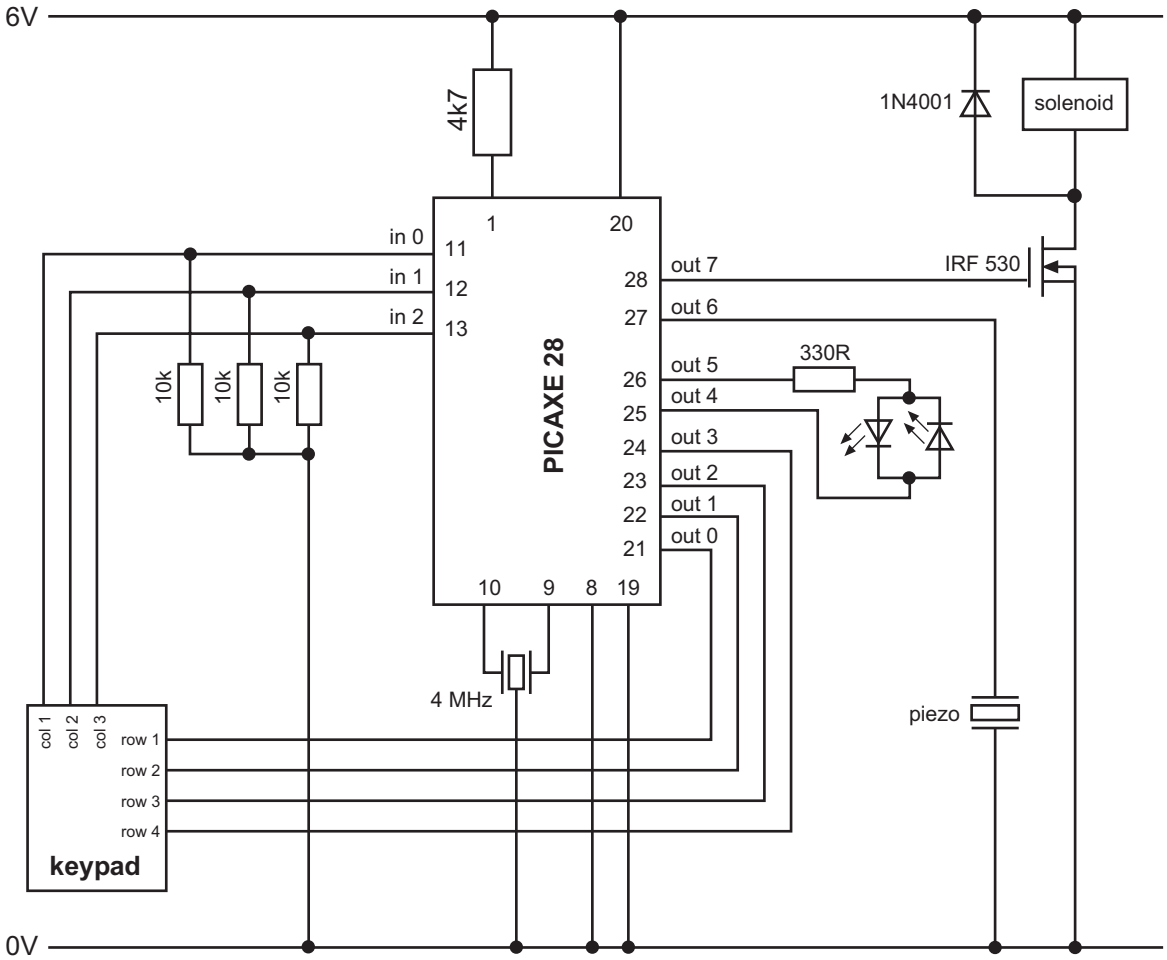
If all the output lines are switched off (at logic level 0) and you press a switch, the input signals will still remain at logic level 0. This is because there is no .high. level signal anywhere on the keypad, and so the only possible input signal to the controller is still 0-0-0.

However if you switch one, and only one, output line high, and then press a switch on the corresponding row, a high input signal will be generated. For instance, if you press button 1 when row 1 is switched high, the input signal to the controller will be 1-0-0. As you have switched one, and only one, output row high, the controller therefore knows that the 1-0-0 pattern can only have been produced by switch 1. Switches 4,7, and * could not have produced the signal because their output rows are still low. This means that if you continuously scan the keypad, by switching each individual row high in turn, you can uniquely identify each single switch.

Program Explanation

The program is fairly complex, and exists around two variables. The first variable, key_pos, is used to remember if is the first, second, third or forth push in the secret number (9-3-5-1) sequence. If a number is got wrong this variable is reset back to 0 to start again.

The second variable, key_value, is used to store the value of any key pushed. As each row of the keypad is scanned, this value is 'pre-loaded' with the value 0,3,6 or 9. If a key in that row is pushed, another value 1,2 or 3 is added to this pre-loaded value. Therefore when a key is pressed a unique value will be left in the key_value. This unique value can then be checked against the code to make sure it was the correct switch that was pushed.



  Revolution Education Ltd. Email: info@rev-ed.co.uk Web: www.rev-ed.co.uk

Program Listing

```
‘ Keypad Lock
‘ For PICAXE-28

‘output 7 - FET to drive solenoid bolt
‘output 6 - piezo sounder
‘output 4,5 - bicolour LED
‘output 3 - row 4
‘output 2 - row 3
‘output 1 - row 2
‘output 0 - row 1

‘input 0 - column 1
‘input 1 - column 2
‘input 2 - column 3


symbol key_pos = b0        ‘ number of keys pressed
symbol key_value = b1      ‘ value of key pressed



‘ *** reset key position to 0 ***

init:
     let key_pos = 0

‘ *** now scan each row in turn ***
‘ *** by setting only 1 row (and LED) high ***
‘ *** if a switch is hit jump down to score section below ***

scan:
‘ scan row 1
     let key_value = 0
     let pins = %00100001
     if pin0 = 1 then add3
     if pin1 = 1 then add2
     if pin2 = 1 then add1

‘scan row 2
     let key_value = 3
     let pins = %00100010
     if pin0 = 1 then add3
     if pin1 = 1 then add2
     if pin2 = 1 then add1

‘scan row 3
     let key_value = 6
     let pins = %00100100
     if pin0 = 1 then add3
     if pin1 = 1 then add2
     if pin2 = 1 then add1

‘scan row 4
     let key_value = 9
     let pins = %00101000
     if pin0 = 1 then add3
     if pin1 = 1 then add2
     if pin2 = 1 then add1

     goto scan
```



```
‘ *** Score section. ***
‘ *** key value will already be 0, 3, 6, or 9 ***
‘ *** so add 1, 2 or 3 to this value ***

add3: let key_value = key_value + 1
add2: let key_value = key_value + 1
add1: let key_value = key_value + 1

‘ *** Make a beep ***

     sound 6,(60,50)

‘ *** Now increase position by 1 ***
‘ *** and test for 1st, 2nd 3rd or 4th push ***

     let key_pos = key_pos + 1

     if key_pos = 1 then test1
     if key_pos = 2 then test2
     if key_pos = 3 then test3
     if key_pos = 4 then test4
     goto init

‘ *** Now test the value for each position individually ***
‘ *** If value is wrong restart, if correct loop until ***
‘ *** fourth go. If forth is correct open lock! ***

‘ *** Key code is set to 9-3-5-1 ***

test1:    if key_value = 9 then scan
     goto init

test2:    if key_value = 3 then scan
     goto init

test3:    if key_value = 5 then scan
     goto init

test4:    if key_value = 1 then open
     goto init

‘ *** Got here so open lock and set LED green for 5 sec ***

open: let pins = %10010000
     wait 5
     goto init
```

