ROB-GY 5103 Mechatronics
Assignment 2

Aditya Wagh (amw9425)
Naveen Kumar (nk3090)
Gordon Oboh (goo2006)

October 2021

Contents

1 Project 1

1.1 Problem

Using BS2, appropriately-sized resistors, LEDs, and buttons, create a circuitry that interfaces two buttons to BS2 pins P0 and P1 and two LEDs to BS2 pins P4 and P5. Next, write a PBasic program that queries the status of two buttons "simultaneously" and activates the corresponding LEDs (B1→LED1, B2→LED2, etc.) simultaneously. Note: Inactive button → LED off and active button → LED on. **Caution:** Make sure to use appropriate resistors to limit current from/into BS2.

1.2 Solution

We have declared pin 0 and 1 of basic stamp 2 as input pins, in which the state of the button is the input data. Then we declare pin 4 and pin 5 as output pins according to the circuit diagram. There are 4 cases running under an indefinite loop. When the input 0 and 1 both are active, in other words, both buttons are pressed, then we'll receive output on the 4th and 5th pin. Similarly, in other cases LEDs will get output according to the corresponding input button pressed. Button 0 will cause LED 4 to glow and Button 1 will cause LED 5 to glow.

1.3 Implementation

```
' {£STAMP BS2}
' {£PBASIC 2.5}

' Initializing input pins
INPUT 0 ' Input pin for button 1
INPUT 1 ' Input pin for button 2

' Initializing output pin
OUTPUT 4 ' Output pin for LED 1
OUTPUT 5 ' Output pin for LED 2

DO
  IF IN0 = 0 AND IN1 = 0 THEN        ' Button 1 and Button 2 are pressed
    OUT4 = 1 ' LED 1 - turned on
    OUT5 = 1 ' LED 2 - turned on
  ENDIF

  IF IN0 = 0 AND IN1 = 1 THEN        ' Button 1 is pressed, Button 2 is not pressed
    OUT4 = 1 ' LED 1 - turned on
    OUT5 = 0 ' LED 2 - turned off
  ENDIF

  IF IN0 = 1 AND IN1 = 0 THEN        ' Button 1 is not pressed, Button 2 is pressed'
    OUT4 = 0 ' LED 1 - turned off
    OUT5 = 1 ' LED 2 - turned on
  ENDIF

  IF IN0 = 1 AND IN1 = 1 THEN        ' Button 1 and Button 2 are not pressed
    OUT4 = 0 ' LED 1 - turned off
    OUT5 = 0 ' LED 2 - turned off
  ENDIF

LOOP
```

1.4 Images & Demo

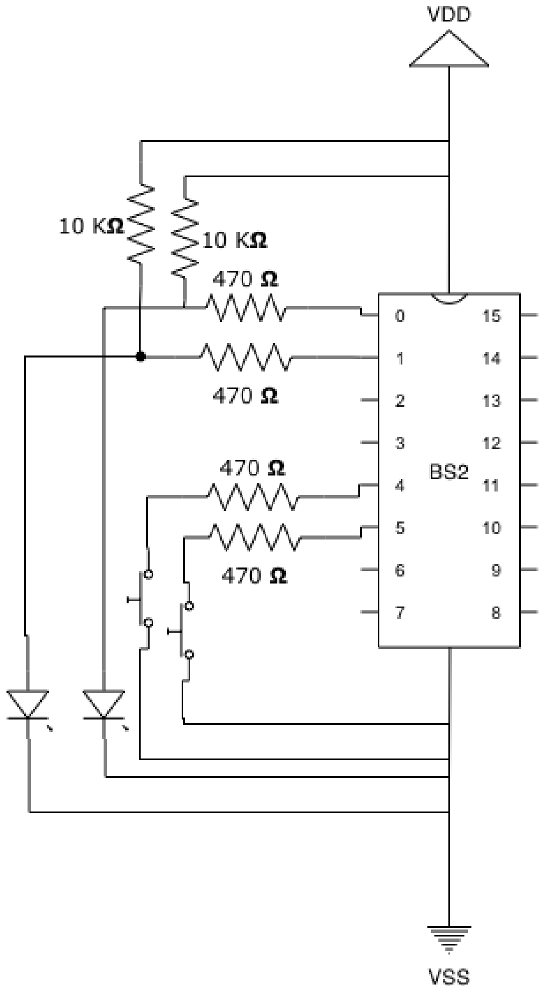

Figure 1: Circuit for Project 1

2 Project 2

2.1 Problem

Cold and Hot Game A player guesses a number between 0 and 99. Each time the player enters a guess; s/he will get feedback in the form of cold (too low), hot (too high), or win (correct guess). Three LEDs and two buttons will be used. The LEDs stand for the feedback (one for cold, one for win, and one for hot). A number between 0 and 99 can be selected using two buttons the same way as in the hint below. Also, the guesses and the feedback must be displayed on the debug window. The player has three chances to guess the correct answer. When the game is over, turn on all the LEDs. Hint: Let the two buttons be denoted as B1 and B2. Using B1 and B2, a user can input a number from 0 to 99. Specifically, the button B1 is used to input the tens digit and the button B2 is used to input the ones digit. Thus, e.g., by pressing B2 three times and B1 two times, the user enters the number 23.

2.2 Solution

2.3 Implementation

2.4 Images & Demo

3 Project 3

3.1 Problem

RCTime Measurement: Construct a series RC circuit. For "R" use a 5K ω resistor. For "C" you are to use (i) 0.1 μ F, (2) 1 μ F, and (3) 10μ F. using BS2, appropriate circuitry, and PBasic program, determine the RCtime value returned by the PBasic command "RCTIME" for the three series RC-circuit. Analytically compute the RCtime value that ought to be returned by BS2. Comment if there are any discrepancies between analytical and experimental values and ascribe the reasons for same.

3.2 Solution

For Problem 3, we have to measure the RC Time to let the capacitor discharge from 5 V to 1.4 V. In the program, we declared a variable "Pot" which will provide the result. First, we provide a high output on pin 0 and in second line we set the discharge time to 100 milliseconds. Then the RCTime function will measure the time for the capacitor to discharge from 5V to 1.4V. The units are in 2 micro second. So, we have to multiply with 2 to get the real result from the micro-controller.

Time in μ F	0.1μ F	1μ F	10μ F
Experimental Reading	580	5942	63620
Analytical Reading	636	6365	636650

Table 1: RC time measurements

In analytical values, the calculation will be like RCTime = R*C*1.273, 1.273 comes from ln(1.4 / 5) = ln(Vf/ Vs) There is some difference between the results we received from the Readings from Basic stamp and Analytical Calculations. These differences can arise due to errors in resistors and capacitors, some internal processes like analog to digital conversion, delays in execution of commands.

3.3 Implementation

```
'{£STAMP BS2}
'{£PBASIC 2.5}

'Initializing variables
Pot             VAR Word             ' Store the value of the time constant

'Initializing input pins
INPUT               IN0

Calculate_RCtime:
    HIGH            0
    PAUSE           100
    RCTIME          0, 1, Pot
    DEBUG           DEC ? Pot
    PAUSE           500
    END
```

3.4 Images & Demo

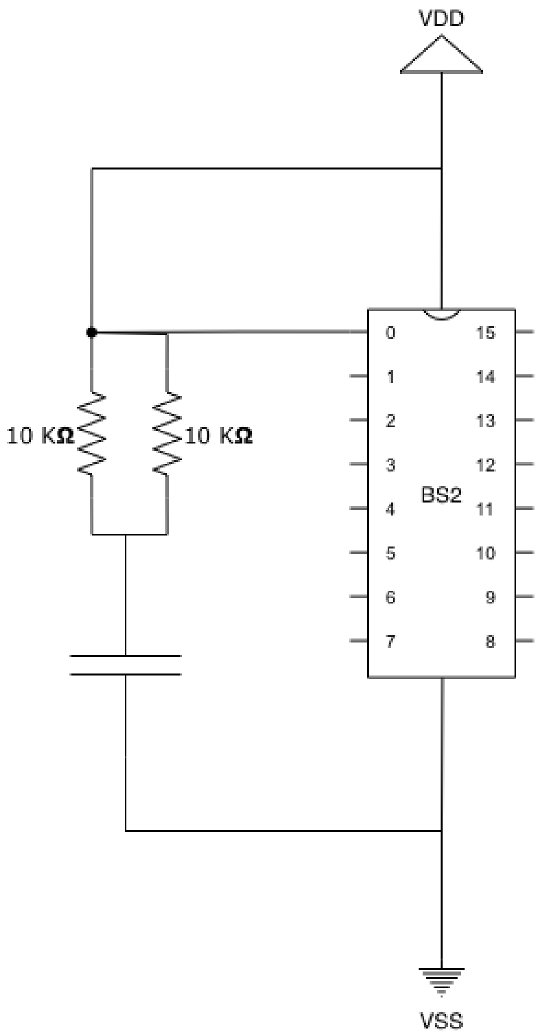

Figure 2: Circuit for Project 3

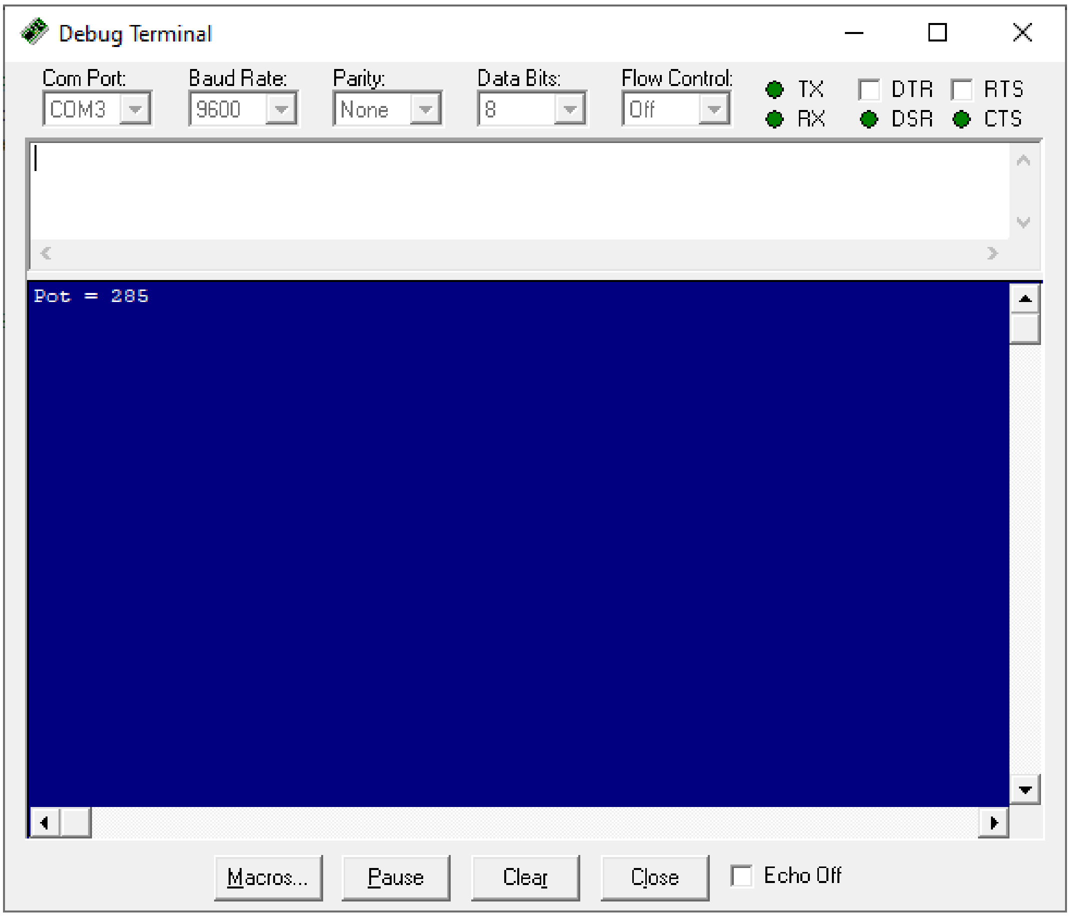

Figure 3: Output for 0.1 μF capacitor

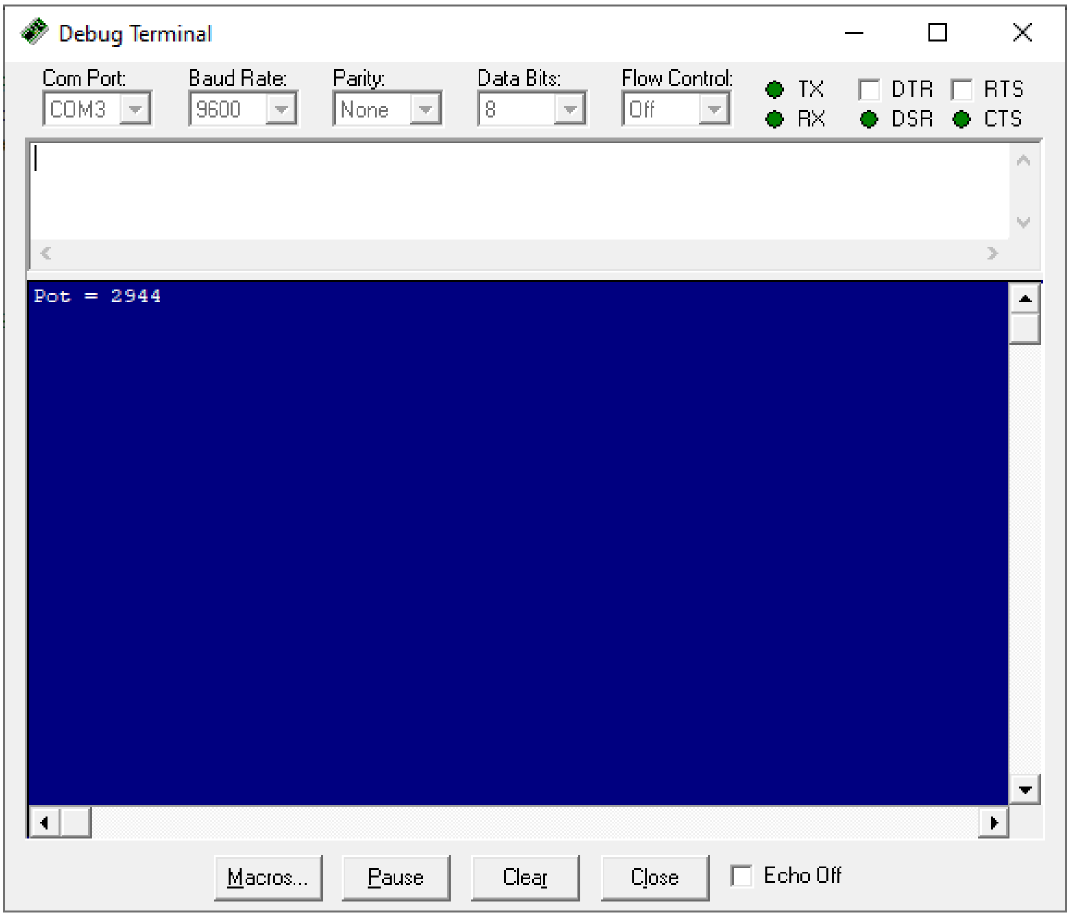

Figure 4: Output for 1 μF capacitor

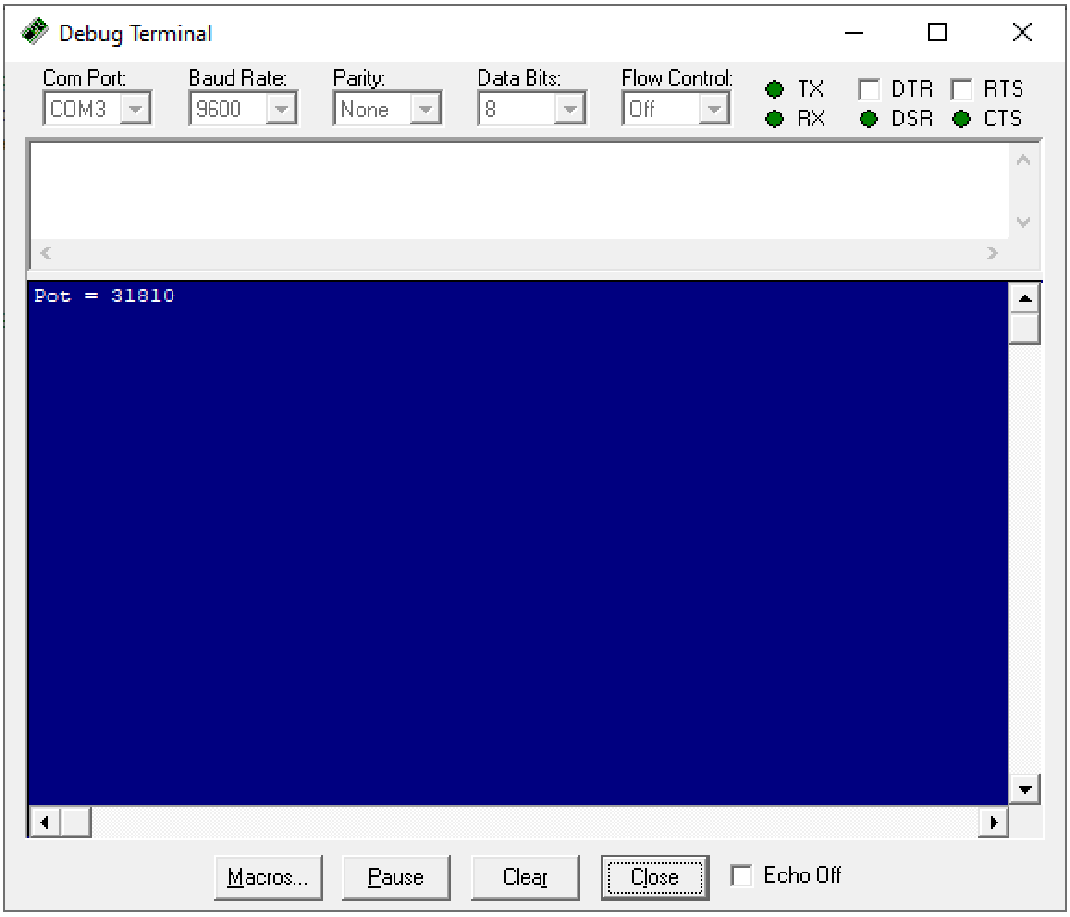

Figure 5: Output for 10 μF capacitor

4 Project 4

4.1 Problem

Chapter 4 of Parallax's Basic Analog and Digital Manual considers digital to analog (D2A) conversion using a resistive ladder circuit. Specifically, Fig 4.1 provides the schematic for a 4-bit D2A converter. Compare this schematic with the one given in Lecture Topic 4 notes (Lecture # 5, slide # 34). The one in the notes is a correct implementation. Now, implement the 4-bit D2A converter and record voltage output produced for all 16 possible states of 4-bit D2A converter. Also perform analytical computation to obtain voltage output produced when only one bit at a time is turned on (others off). That is obtain voltage output for b0=1, (b1=b2=b3=0); b1=1, (b0=b2=b3=0); b2=1, (b0=b1=b3=0); and b3=1, (b0=b1=b2=0).

4.2 Solution

We are referring to the diagram given in Chapter 4 of Parallax's Basic Analog and Digital Manual Fig. 4.1 and Lecture Topic 4 notes (Lecture # 5, slide # 34). When we compare both the circuits given, we found that we are receiving 5 voltage a peak in Lecture # 5, slide # 34 and 3.33 in Fig. 4.1 of the manual. We implemented both and the results are as presented in the screenshot.

In this problem, we are using a resistive ladder circuit. Which is receiving inputs from 4 digital i/o pins of Basic Stamp 2 and the resultant output is going to ADC0831 which is measuring the voltage of the output and after converting it into digital output, it sends it to BS2 for printing output on the screen.

4.3 Implementation

```
' {£STAMP BS2}
' {£PBASIC 2.5}

' Declarations
adcBits         VAR Byte
v               VAR Byte
r               VAR Byte
v2              VAR Byte
v3              VAR Byte
n               VAR Nib

' Initialization
CS              PIN 0
CLK             PIN 1                                   ' Clock
DataOutput      PIN 2                                   ' Pin for the output of ADC

'Start display
DEBUG           CLS, "DAC Nibble Values", CR
DEBUG           "Decimal Binary DVM", CR

' Main Routine
FOR n = 0 TO 15
    GOSUB       DAC
    GOSUB       ADC_Data
    GOSUB       Calc_Volts
    GOSUB       Display
NEXT
STOP

' Subroutines
DAC:
    OUTPUT      7
```

```
    OUTPUT      6
    OUTPUT      5
    OUTPUT      4
    OUT7 = n.BIT3                                   ' MSB of DAC output
    OUT6 = n.BIT2
    OUT5 = n.BIT1
    OUT4 = n.BIT0                                   ' LSB of DAC output
RETURN

ADC_Data:
    LOW         CLK                                 ' Set to low to activate the ADC
    LOW         CS                                  ' Set to low to activate the ADC
    PULSOUT     CLK, 210
    SHIFTIN     DataOutput, CLK, MSBPOST, [adcBits\8]
    HIGH        CS
RETURN

Calc_Volts:
    v = 5 * adcBits / 255
    r = 5 * adcBits // 255
    v2 = 100 * R / 255
    v3 = 100 * R // 255
    v3 = 10 * v3 / 255
    IF (v3 >= 5) THEN v2 = v2 + 1
    IF (v2 >= 100) THEN
    v = v + 1
    v2 = 0
    ENDIF
RETURN

Display:
        DEBUG       DEC2 n, "       ", BIN4 n, "  "
        DEBUG       DEC1 v, ".", DEC2 v2, " Volts", CR
RETURN
```

4.4 Images & Demo

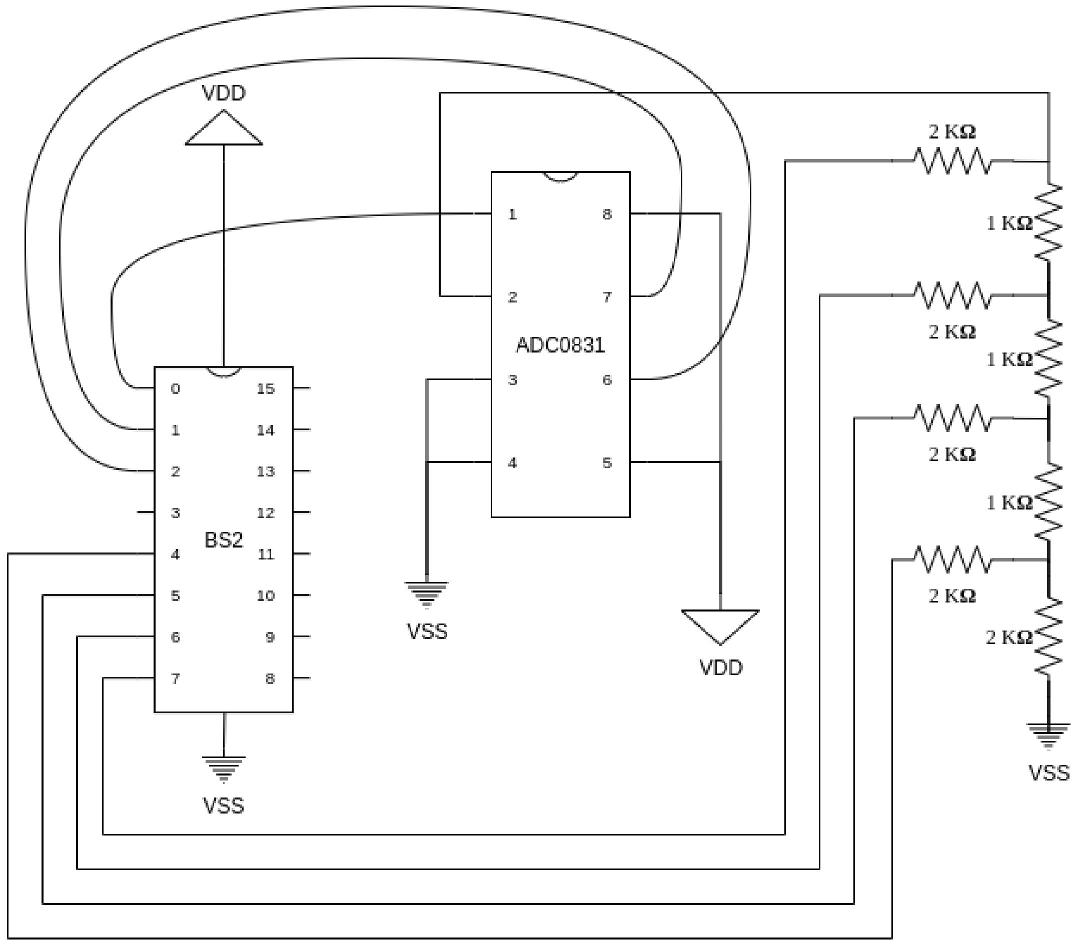

Figure 6: Circuit for Project 4