Frequency Generator

TEAM (11):

MAZEN AHMED RAMZY 6999

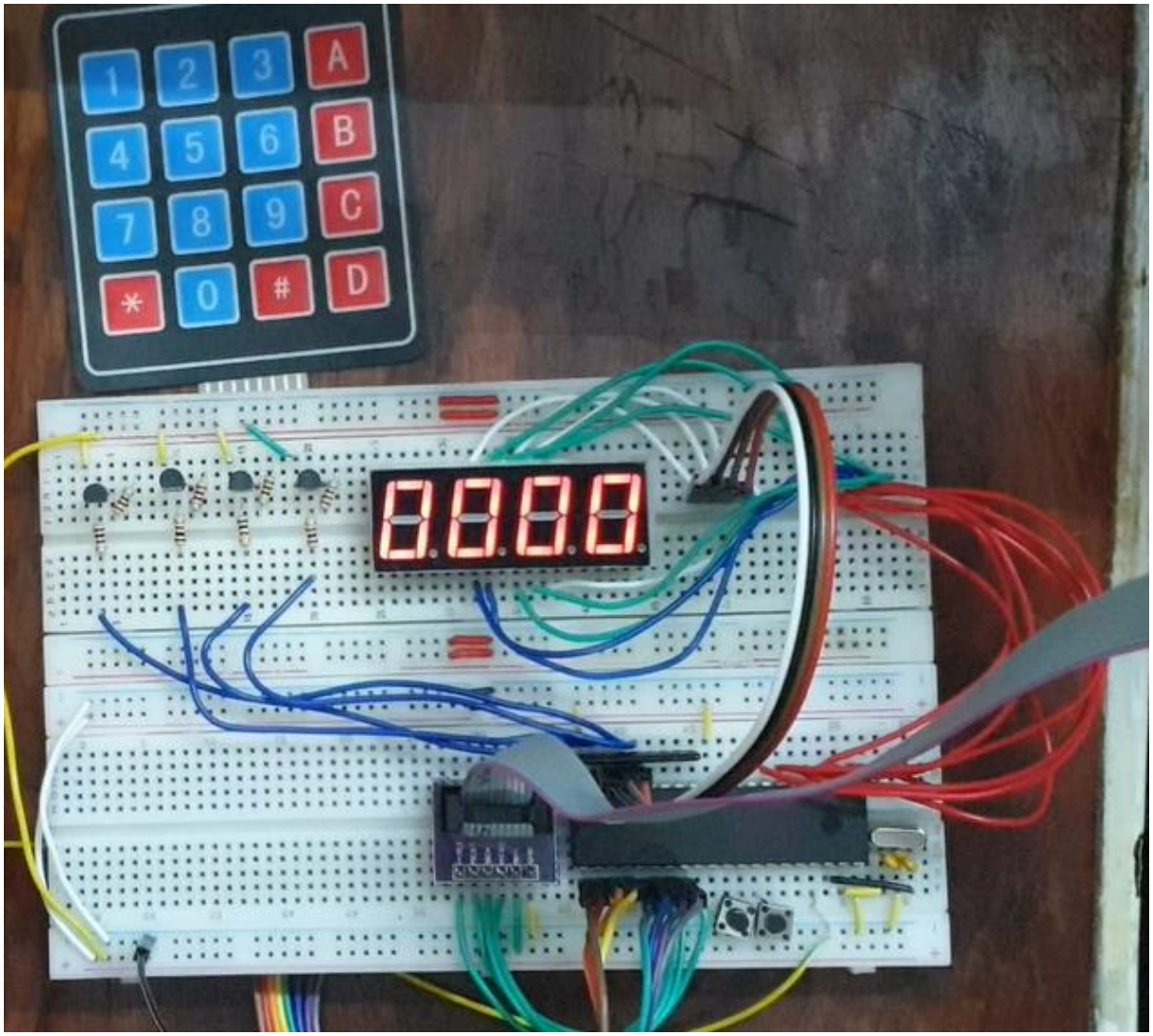
BAHAAELDIN MOHAMED GAMALELDIN 6726

ZIYAD MOHAMED AL SAEED 6474

ABDALLA FAHMY KASSEM 6866

MAHMOUD SAMI 7114

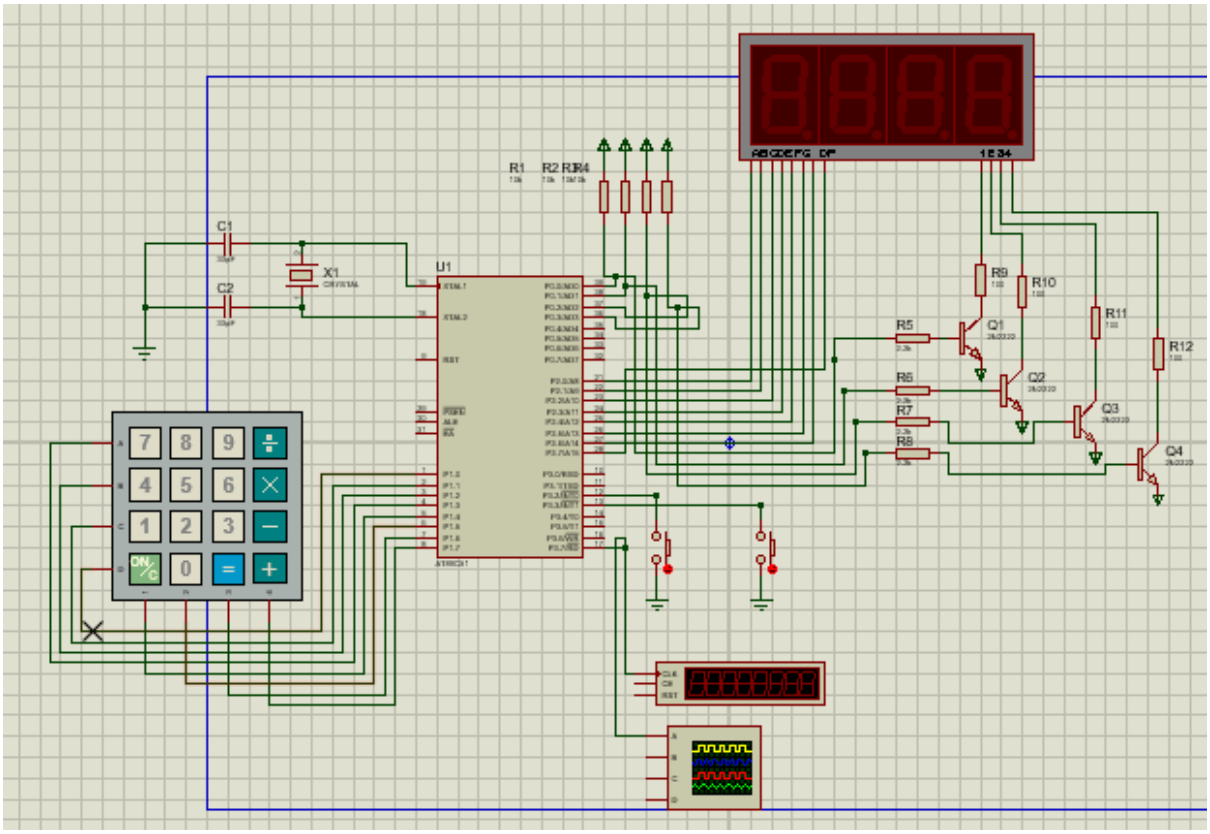
Hardware Connection



Components used:

- 4x4 keypad
- Transistors (2N3906)
- 4 digit 7-segment
- 2 push buttons
- Crystal: 11.0592 MHz

Proteus simulation connection



Connection description:

- Keypad connected to port 1
- Digits of 7-segment connected to port 0 with pull resistors and transistors
- Segments (a,b,c,d,e,f,g) is connected to port 2
- Frequency count connected to pin 3.7
- Push buttons connected to port 3.2 and 3.3

How does the circuit work?

First, the user enters the desired frequency using the keypad up to 9999.

After that, the user presses on the interrupt 0 button (pin 3.2).

The desired frequency is generated at pin 3.7

To stop the frequency generation. User presses on the interrupt located at button pin 3.3 and the user can enter a new desired frequency

Code explanation:

The program is written in C for the AT89S51 microprocessor. The code takes input from a 4x4 keypad and generates a square wave with the desired frequency on an output pin.

The code initializes the 7-segment display, common anode pins for the digits, row and column pins for the keypad, and an interrupt pin. It also defines some functions for delay, setting the wave, and managing the timer.

The main function initializes the pins and sets up the interrupt and timer. It then enters an infinite while loop where it scans the keypad for input. Once the user has entered a frequency, the program calculates the timer value required for the desired frequency and sets the timer to generate the square wave. The square wave is output on the FREQ_OUT pin. (3.7)

The program first defines the pins for the segments of the 7-segment display, the common anode for the digits, and the row and column pins for the keypad as special function registers (sbit). It then defines a few variables for use in the program: segconv, freq, temp, seg_table, incseg, hold, and key, used to manage the output frequency and the seg_table array which is displayed on the 4-digit 7-segment ***(explained in the video).***

The program then initializes the pins for the segments and digits as output and sets up the external interrupt and timer interrupt. It also sets the display to show a '0' initially.

The while loop in the main function scans the keypad for input. It first sets the key variable to a default value which is not from 0 to 9. It then scans each row of the keypad to check if any keys have been pressed. If a key has been pressed, it sets the key variable to the value of the pressed key and sets the hold variable to 1 to prevent multiple presses.

Once the user has entered a frequency, the program calculates the timer value required for the desired frequency and sets the timer to generate the square wave. The square wave is output on the FREQ_OUT pin.(3.7).

The program uses the T1Delay function to create a delay of a specified time. It also uses the segments_init function to initialize the pins for the 7-segment display. The get_key function scans the keypad and returns the pressed key value. The setwave function sets the FREQ_OUT pin to output a square wave with a specified frequency. The settimer function sets the timer to generate a square wave with a specified frequency. The starttimer function starts the timer. The resettimer function resets the timer. The stoptimeINT function stops the timer interrupt. The stoptimer function stops the timer.