Logo

Description automatically generated A picture containing text, clipart

Description automatically generated

**Microcontrollers and Embedded Systems**

**MTS- 311**

**Project Report**

**Group Members:**

NC Fatema Asif

NC Zunnash Khan

NC Noor ul Huda Rasool

Date: 5 January 2022

Submitted to:

Sir Qasim

# Introduction

The objective of this project was to create a generic board based on AT89C51 and to program it based on or desired outcome for different modes.

# Hardware

## Board 1

The AT89C51 board was made of the following components:

|  |  |  |
| --- | --- | --- |
| Component |  | Function |
| 74LS245 Tri-State Buffer ICs | arduino component :: SN74LS245N DM74LS245N 74LS245 Octal Bus Transceiver -  Buy Quality Electronics | Hallroad Lahore | The IC is used as a buffer between the microcontroller ports and the LED Bargraphs to prevent the current to exceed a minimum threshold thus preventing the burning of ports. |
| LED Bargraphs | A picture containing text  Description automatically generated | These show the state of the ports on the microcontroller. |
| Capacitors |  | For smoothening in the ziff circuit. |
| Resistors |  | Used to reduce current flow. |
| Resistor pack (10k & 1k Ohms) | [This Photo](http://electronics.stackexchange.com/questions/149549/were-single-in-line-packages-sip-ever-popular-in-the-west-or-even-used-much) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/) | Used to pull up port 0 and pull down the rest of the ports. |
| Reset button |  | Used in the reset circuitry for resetting the microcontroller. |
| Headers | A picture containing electronics  Description automatically generated | To take inputs from an external dip switch board. |
| AT89C51 Microcontroller | A picture containing electronics, circuit  Description automatically generated | The central component of the board. Used to execute the program code. |
| Oscillator Crystal 11.0592 MHz | Text  Description automatically generated | To provide a frequency to the microcontroller. |

The process to make the PCB board involved the following:

1. Etching
2. Drilling
3. Soldering

The PCB layout for our project was as below:

Schematic

Description automatically generated with medium confidence

The schematic was as follows:

A picture containing chart

Description automatically generated

The final board looked as follows:

A close-up of a circuit board

Description automatically generated with medium confidence

## Board 2:

The second board that we created was to take external inputs for port 0 and port 1 through dipswitches. This board was completed with dipswitches, resistor banks, resistors and headers.

It may be seen in the picture below.



# Software

## Mode 1

Mode 1 involved taking a value for t from P1 and solving the distance equation, with a=3, and v=4 and display the answer on P2 and P0 with the higher nibble on P2 and lower nibble on P0.

A proteus simulation of the code can be seen below which takes t = 2 and gives an output of 14.

Chart

Description automatically generated

Next, for larger values, the code worked to show the higher nibble on P2 and lower on P0. For example, a value of 12 from P1 should give (264)10 that corresponds to (0108)16 in hexadecimal. This can be seen in the image below where 01 are represented on P2 and 08 on P0.

Chart

Description automatically generated

This code was as follows:

Mode1:

MOV R0, P1

MOV A, R0

MOV B, A

MUL AB

MOV R1, A

MOV R2, B

MOV B, #2

DIV AB

ADD A, R1

MOV R1, A

MOV A, R2

MOV B, #2

DIV AB

ADDC A, R2

MOV R2, A

MOV A, #4;

MOV B, R0

MUL AB

ADD A, R1

MOV R1,A

MOV A, B

ADDC A, R2

MOV R2, A

JNC SKIP

SETB P3.2

SKIP:

MOV P0, R1;

MOV P2, R2

MOV R5, #4

L1:

MOV R6, #255

L2:

MOV R7, #255

L3:

JB P3.1, EXIT

JB P3.0, EXIT

DJNZ R7, L3

DJNZ R6, L2

DJNZ R5, L1

EXIT:

JMP AGAIN

## Mode 2

For this mode, we took values from 0 to 13 from P0 and output the corresponding value of that index from the Fibonacci Series on port 2.

For example, for an index value of 4, the output shown is 3

Chart

Description automatically generated

The code that did this is shown as follows:

Mode2:

MOV DPTR, #300H

AgainM2:

FS: MOV A, P1

MOVC A, @A + DPTR

MOV P2,A

MOV R5, #9

M1L1: MOV R4, #255

M1L2: MOV R3, #255

M1L3: DJNZ R3, M1L3

DJNZ R4, M1L2

DJNZ R5, M1L1

MOV P2, #0H

SJMP FS

JMP AgainM2

## Mode 3

This mode involved blinking the LEDs of P0 and P2 progressively from MSB to LSB and LSB to MSB respectively, with the speed of the blinking depending on the value taken from P1. Hence, for a higher value closer to 255, the blinking was faster and for a lower one, it was much slower.

Chart

Description automatically generated

The code for this mode is as follows:

Mode3:

AGAINM3:

MOV B, P1

MOV A, #0FFH

SUBB A, B

MOV B, #1H

ADD A, B

MOV P2,#0H

BLINKLEDS:

MOV P2, #1H

MOV P0, #80H

CALL DELAYM3

MOV P2, #2H

MOV P0, #40H

CALL DELAYM3

MOV P2, #4H

MOV P0, #20H

CALL DELAYM3

MOV P2, #8H

MOV P0, #10H

CALL DELAYM3

MOV P2, #10H

MOV P0, #08H

CALL DELAYM3

MOV P2, #20H

MOV P0, #04H

CALL DELAYM3

MOV P2, #40H

MOV P0, #02H

CALL DELAYM3

MOV P2, #80H

MOV P0, #01H

CALL DELAYM3

JMP BLINKLEDS

JMP AGAINM3

## Mode 4

This mode involved the blinking of P0 and P2 simultaneously with a set frequency of 3 Hz. Depending on the input from P1, the duty cycle of the resulting square wave varied.

The image below shows a 50% duty cycle for an input of 128 and the time period for a single cycle is 335 ms.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

The code for this mode is as follows:

Mode4:

AGAINM4:

MOV TMOD, #1H

MOV A, P1

CJNE A, #25, A1

B1:

MOV R7, #1

MOV R5, #11

;ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A1:

JC B1

CJNE A, #50, A2

B2:

MOV R7, #2

MOV R5, #10

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A2:

JC B2

CJNE A, #75, A3

B3:

MOV R7, #3

MOV R5, #9

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A3:

JC B3

CJNE A, #100, A4

B4:

MOV R7, #4

MOV R5, #8

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A4:

JC B4

CJNE A, #125, A5

B5:

MOV R7, #5

MOV R5, #7

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A5:

JC B5

CJNE A, #150, A6

B6:

MOV R7, #6

MOV R5, #6

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A6:

JC B6

CJNE A, #175, A7

B7:

MOV R7, #7

MOV R5, #5

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A7:

JC B7

CJNE A, #200, A8

B8:

MOV R7, #8

MOV R5, #4

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A8:

JC B8

CJNE A, #225, A9

B9:

MOV R7, #9

MOV R5, #3

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A9:

JC B9

CJNE A, #250, A10

B10:

MOV R7, #10

MOV R5, #2

ACALL LOOPON

ACALL LOOPOFF

JMP AGAINM4

A10:

JC B10

CJNE A, #255, A11

B11:

MOV R7, #11

MOV R5, #1

ACALL LOOPON

;ACALL LOOPOFF

JMP AGAINM4

A11:

JC B11

JMP AGAINM4

The functions that were used in the program were as follows:

Mode3:

AGAINM3:

MOV B, P1

MOV A, #0FFH

SUBB A, B

MOV B, #1H

ADD A, B

MOV P2,#0H

BLINKLEDS:

MOV P2, #1H

MOV P0, #80H

CALL DELAYM3

MOV P2, #2H

MOV P0, #40H

CALL DELAYM3

MOV P2, #4H

MOV P0, #20H

CALL DELAYM3

MOV P2, #8H

MOV P0, #10H

CALL DELAYM3

MOV P2, #10H

MOV P0, #08H

CALL DELAYM3

MOV P2, #20H

MOV P0, #04H

CALL DELAYM3

MOV P2, #40H

MOV P0, #02H

CALL DELAYM3

MOV P2, #80H

MOV P0, #01H

CALL DELAYM3

JMP BLINKLEDS

JMP AGAINM3