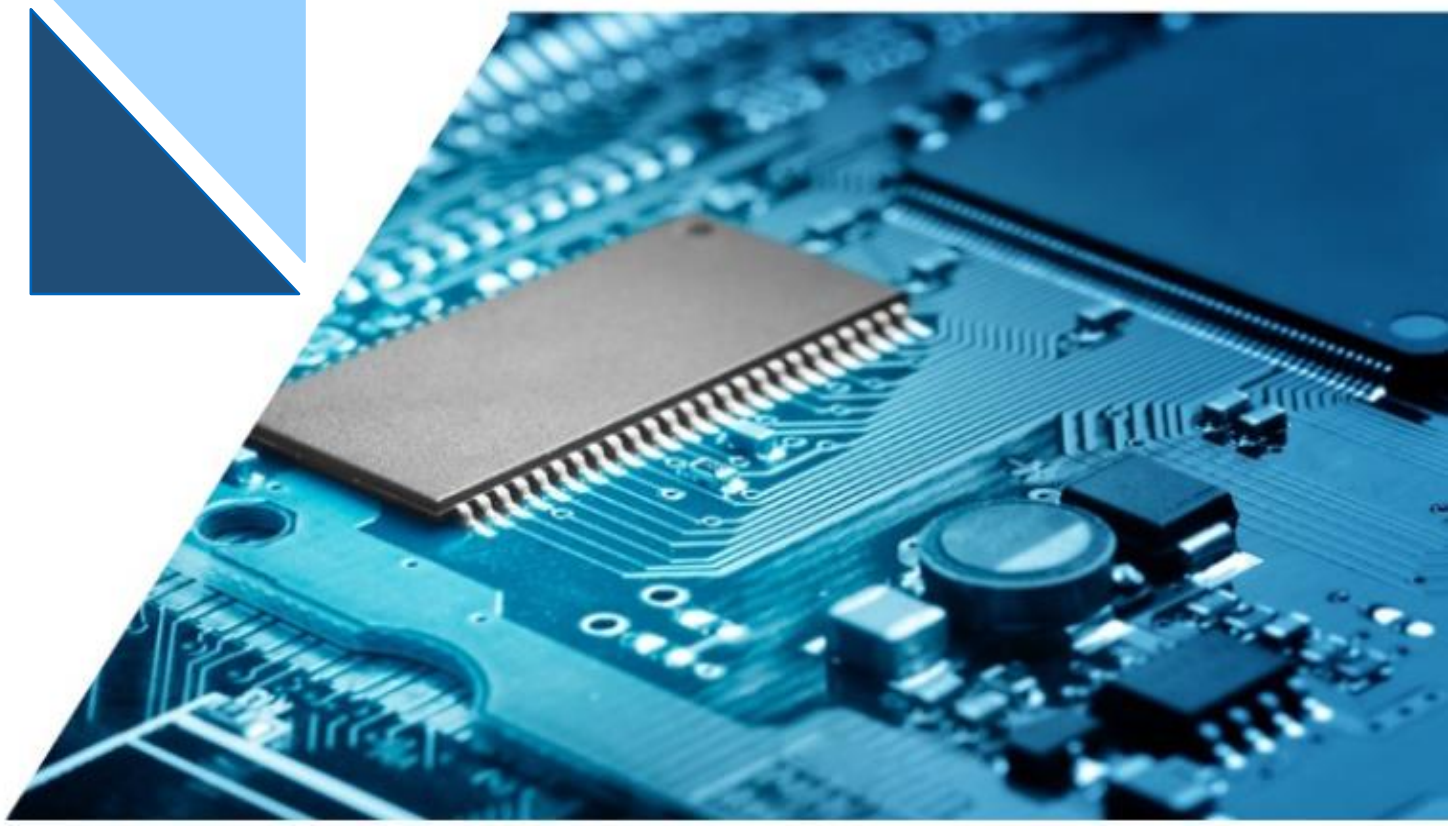
Microprocessor Based System

Project Title:
Temperature controlled DC



Group members

- Rahima Khan (CS-073)
- Aimen ejaz(CS-074)
- Syed Muhammad Imran(CS-102)
- Ammar Darugar(CS-302)

CONTENTS

Introduction:

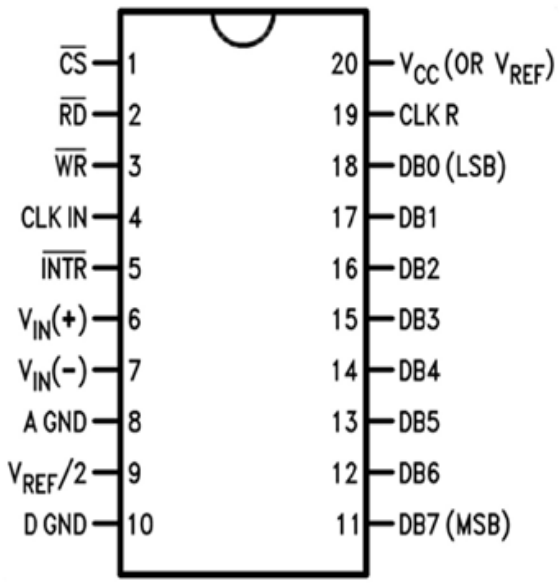
CS-301 “Microprocessor based system” is related to computer architecture. In which we studied 8086, 8088 microprocessor and Atmega162 microcontroller from AVR family. By using this microcontroller, we have made a temperature-controlled fan

Abstract:

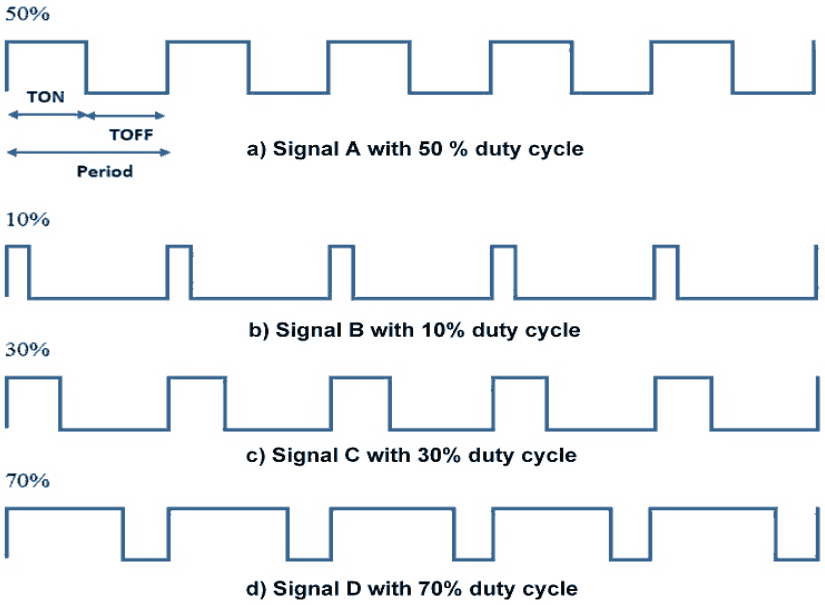
In this project, we made a temperature control dc fan using Atmega162 microcontroller and implement the pulse width modulation technique. When resistance value is changed ADC produce a respective binary stream and send to microcontroller, then according to this value duty cycle set and fan will start rotating according to duty cycle.

Basic theory:

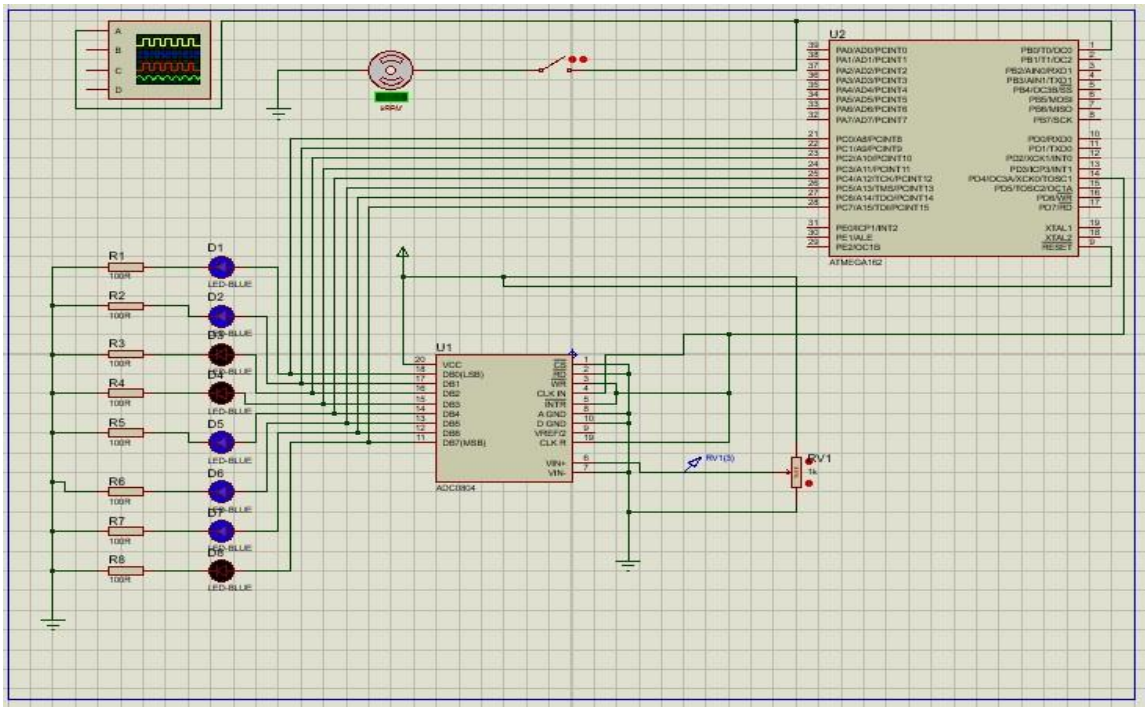
- Analog to digital convertor (ADC):
  ADC is a technique used to convert analog signals to digital data. Here we are going to talk about ADC0804. This is a chip designed to convert analog signal in to 8 bit digital data. This chip is one of the popular series of ADC. As said this chip is specially designed for getting digital data for processing units from analog sources. Its an 8bit conversion unit, so we have 28 values or 1024 values. With a measuring voltage of maximum value 5V, we will have a change for every 4.8mV. Higher the measuring voltage there will be decrease in resolution and accuracy.



- **Pulse width modulation:**
  Pulse Width Modulation (PWM) is a digital signal which is most commonly used in control circuitry. The time during which the signal stays high is called the “on time” and the time during which the signal stays low is called the “off time”.
  The percentage of time in which the PWM signal remains HIGH (on time) is called as duty cycle.


50%
TON
TOFF
Period
a) Signal A with 50 % duty cycle
10%
b) Signal B with 10% duty cycle
30%
c) Signal C with 30% duty cycle
70%
d) Signal D with 70% duty cycle

Simulation:


U2
ATMEGA162
U1
ADC0804
RV1
R1
R2
R3
R4
R5
R6
R7
R8
D1
D2
D5
D6
LED-BLUE

Components required:

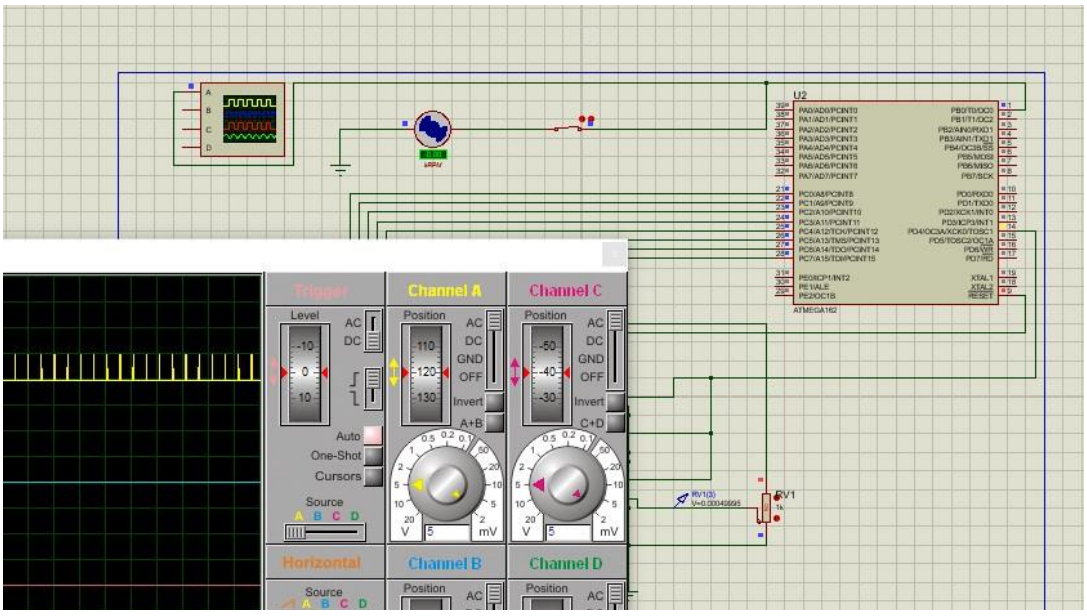
- AVR Atmega162 microcontroller
- ADC0804
- Jumper wires
- DC fan

Working:

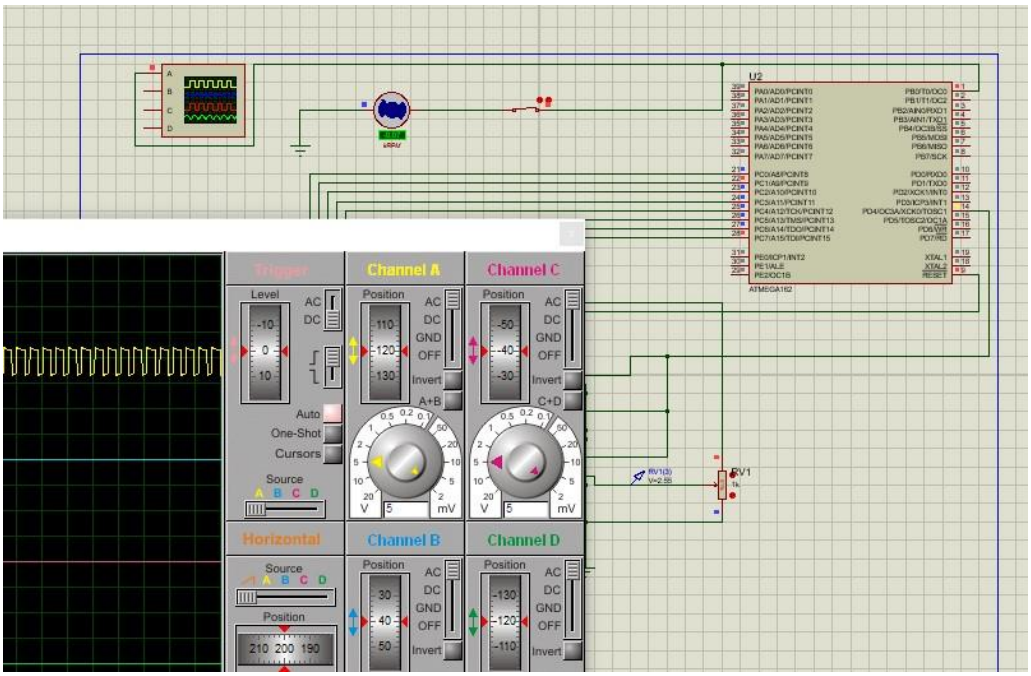
First of all, the clock function call and generate clock signal and send to ADC so that IC will enable. Then adjust the voltage by using potentiometer, this analog voltage will be converted into digital bit stream and send to PINC of microcontroller when another clock signal generate. this bit stream stored in the variable “value”will be compare to predefined range and adjust duty cycle of PWM, according to which fan will start rotating with respect to duty cycle .

Test cases:

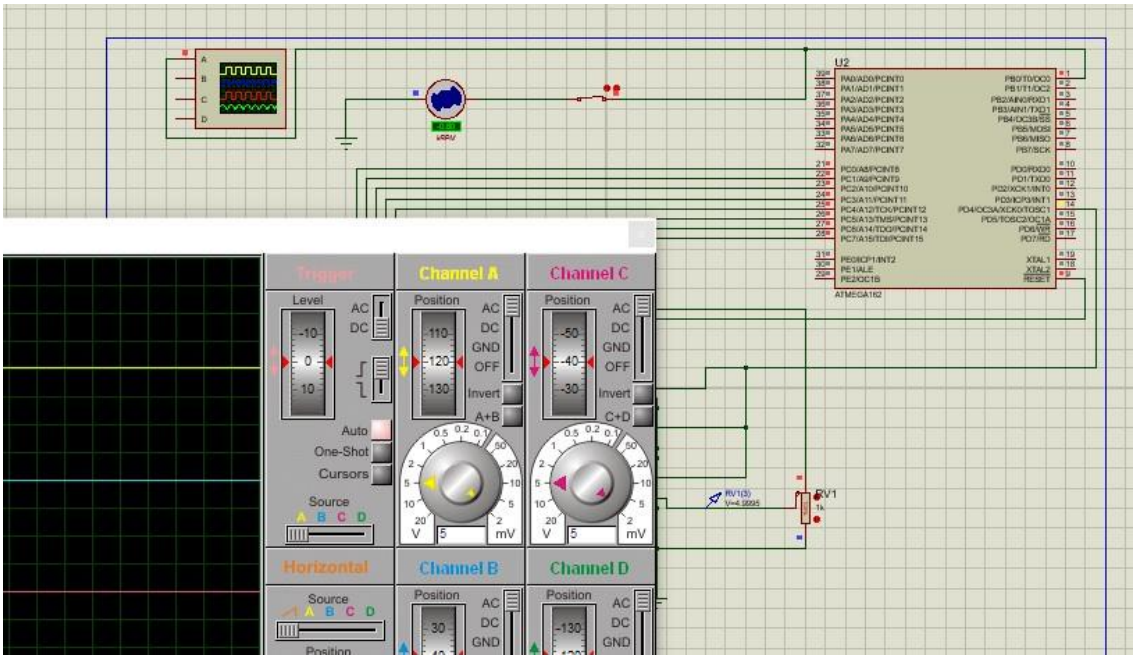
- **Test case 01:**



- **Test case 02:.**



- **Test case:03**



Code:

```
#include <avr/io.h>
#define F_CPU 1000000UL
#define CLOC PD4 //Clock for clock read and clock in of ADC

void CLOC1(void);

int main(void)

{
	DDRC = 0x00; // Pin defined for reading ADC input
	DDRB = 0x01; //Setting pin for OCR0
	DDRD|= (1<<CLOC); //Setting Clock pin
	TCCR0|= (1<<WGM00)|(0<<WGM01)|(1<<CS00)|(1<<COM01)|(0<<COM00); //PWM setting output
	TCCR1B = (1<<CS12)|((1<<CS10));
	TCNT1 = 0;	//counter pin
    char value;    // Variable for reading for ADC
    while (1)
    {

		CLOC1(); // giving clock to ADC for reading values
		value = PINC; // saving readings

		//comparing values for ranges
		if (value<=64)
		{
			OCR0 = 10;
		}
		else if ((value>64)&&(value<=128))
		{
			OCR0 = 80;
		}
		else if ((value>128)&&(value<=193))
		{
			OCR0 = 160;
		}
		else{
			OCR0 = 255;
			}

	}
}

// function of clock
void CLOC1(void){
	if (TCNT1>=50)
	{
		PORTD^=(1<<CLOC);
		TCNT1 = 0;
	}
}
```