DOI: 10.1515/SBEEF-2017-0033

ADVANCED FAN CONTROLLER USING ATMEGA 1284P

F.G. APOSTOL, G. PREDUSCA
Department of Electronics, Telecommunications and Energy Engineering, Valahia University of Targoviste
E-mail: apostolfloringiorgian995@gmail.com, gpredusca@gmail.com

Abstract**. The project aims at creating a monitoring and control circuit in a desktop or complex system with the possibility to set the operating parameters. The project allows automatic and manual control of a certain number of fans using Pulse Width Modulation (PWM) control by the user or automatically by the system monitoring thresholds of minimum and average temperatures (also set by the user within the code). The system was designed with the following options: audible and visual alert when reaching maximum temperatures; time display, PWM percentage and three temperatures shown on a 16x2 LCD; the system-acquired data is then transmitted to a website. Additional features include the possibility of setting a color inside the system, also used as an alert as well as a color adjustment function by sound (for a musical lightshow). The mode of operation is based on the source code written in the Arduino IDE 1.6.3 software with the help of free libraries and free codes provided on the internet.

Keywords: fan, PWM, Atmega 1284P, Arduino IDE.

1. INTRODUCTION

This data acquisition system is designed to work in two ways: manually and automatically. In manual mode, by operating a switch in the ON position it reads three potentiometers that create a voltage whose value is read by the microprocessor (MCU) and converted to a PWM value (0-255) between 0V-5V; this value will operate three digital switches (in the present case three transistors), connected to the collector and a fan pin, the emitter at the GND, and on its base the PWM signal; after this step, the signal is converted into percentages and displayed on both the LCD and the serial transmission. Also in "manual" mode the MCU reads three temperature sensors on a single data bus (each having its own addressing address) and displays the temperature on the LCD and serial; if the maximum temperature is reached, a red LED (or a multi-LED light) will start flashing with a buzzer (which can be turned off) and an error message will be displayed on the LCD and serial up until the temperature drops below the set maximum. Also a Real Time Clock circuit (RTC) will provide the MCU date and time; the data will be displayed on the LCD and the serial with its own power source not depending on the power source of the circuit. All of these processes run at the same time with the possibility of PWM change, temperature reading, monitoring and high temperature warning.

In automatic mode, when the switch is set to OFF, the MCU reads the three temperature sensors and with the help of a minimum and maximum value, it compares the value read for the first time with the minimum value; if this is below the minimum value, the MCU will generate a PWM with the value of 0 meaning the transistor is not actuated and it will keep the supply of the fan interrupted; the LCD and the serial will display a message of the form "no fan is spinning. System OK" (will do for each temperature and fan sensor separately); if the minimum value is exceeded but the maximum value is not, a PWM is generated with a fill value dependent of the read temperature and generated on the base of the transistor tasked with the specific temperature sensor; and in this case a message will be displayed on the LCD and on the serial saying that the fans are being turned on and the values of the generated PWMs. If the maximum value is exceeded, the MCU will directly generate 100% PWM (255) along with the high temperature alert message on the LCD and the serial, as well as the visual alert through an LED, and a buzzer sound [1-4].

We also have a control function for an RGB LED that: by pressing a button you can set a color for the system using the same potentiometers; by pressing another button you can set the lightshow (by type of music) by which the LEDs are lit according to a sound sensor in the background. Note that even if the RGB LED lights up on music, the audible alert and the red LED light will override this function if the temperature limit is exceeded.

In this system I did not use all pins of the microcontroller with enough pins to give me the possibility to update for more functions.

2. PRINCIPLE OF FUNCTIONING

2.1 Auto mode

This system has the main role of monitoring the temperature using sensors with values given by a number of three components in a complex system (example a desktop) whose values are known to vary depending on the usage and the task it performs and whose temperature has to be dissipated by means of radiators and fans, and further transported out of the system.

Figure 1 shows the optimal flow of air in a desktop environment; cold air is taken from the front of the case and below it, and passed through the main components, which are well known for temperature dissipation, these being:

- Central Processing Unit (CPU, 20C-70C);
- Hard Disks (Mechanical between components causes friction, which is why the need for temperature dissipation) (29C-40C);
- Graphics Processing Unit (GPU) (36C-90C);
- RAM (55C-80C);
- Optional: SSD, sound card (slotted).

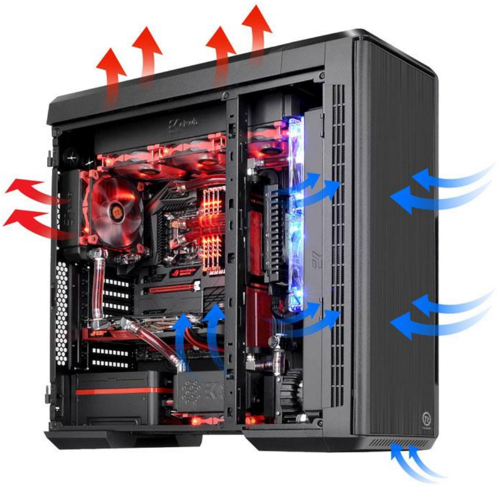


Figure 1. Optimal airflow.

Based on the readings of the sensors, the MCU decides to increase or decrease the speed of certain fans in order to improve air circulation in the area where a higher temperature than normal operation has been detected.

If the temperature value exceeds a certain maximum threshold, the system can alert the user by: audible alert (a buzzer, Figure 6), visual through a red LED or in the case with a housing illumination system, as well as a warning display.

2.2 Manual mode

This mode is different from auto mode, the manual mode leaves the user based on the information provided visually from the display and by his own preference, to determine the fan speeds, regardless of whether the temperature is high or low.

It should also be noted that the high temperature alert is still in operation, except that it will not start the fans any longer, and the decision is placed on the user.

2.3 RGB control

This control makes it possible to display a fairly wide variety of colors using only a mixture of three colors: Red, Green and Blue.

Each of these colors has a intensity value from 0 to 255. So as an example of an intense color of Yellow, we need Red and Green. With an RGB led or an RGB led band, we can display a wide variety of colors inside the system with the main role of aspect and alert.

The intensity of each color is determined by the same method as the fans, a PWM value of 0 to 255.

2.4 Sending the collected data on a web page

Sending data from the monitoring and control system can be done by several methods: Bluetooth, Wifi, GSM, Ethernet or by the method used in this system, on serial (USB).

Using a java-written program through which data is retrieved by Java (Figure 2), interpreted and forwarded by simply concatenating the data with an URL Link, the project made on Dweet.io, and with the help of the freeboard site, this data is retrieved from Dweet.io (Figure 9 and Figure 10).

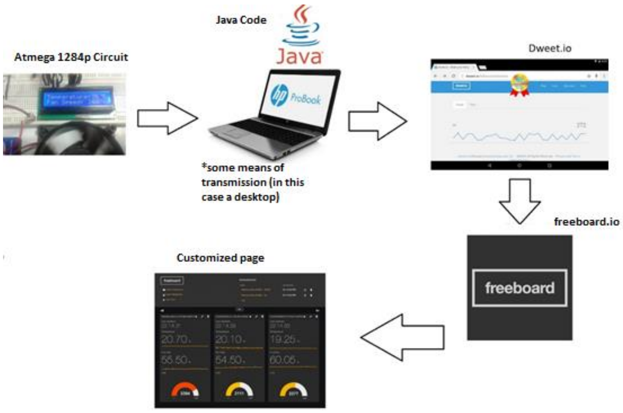
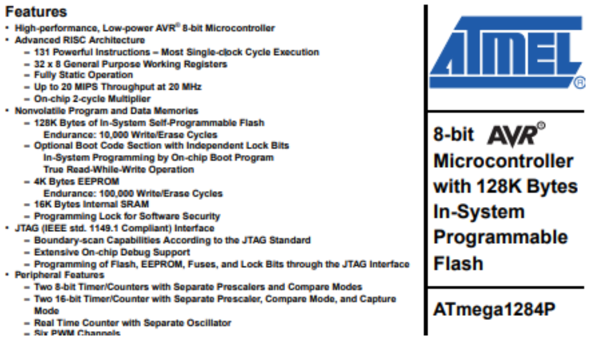


Figure 2. How the data is send to a web site.

3. COMPONENTS USED

3.1 Microprocessor (MCU)

The main reasons for choosing MCU Atmega 1284P (Figure 3) were the high number of pins (40 pins) with 32 I/O (Figure 4) and the large capacity 128Kb Flash memory required for the current program v19.3 and with the possibility of updating when several functions are implemented [5-8].



Features

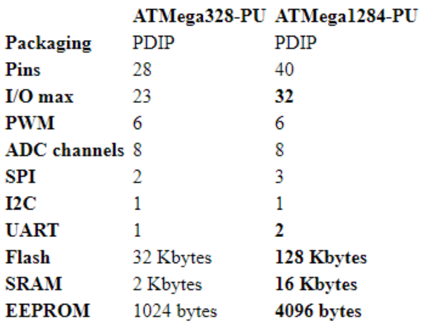
- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 16K Bytes Internal SRAM
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels

ATMEL®

8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

ATmega1284P

Figure 3. Atmega 1284p features.



	ATMega328-PU	ATMega1284-PU
Packaging	PDIP	PDIP
Pins	28	40
I/O max	23	**32**
PWM	6	6
ADC channels	8	8
SPI	2	3
I2C	1	1
UART	1	**2**
Flash	32 Kbytes	**128 Kbytes**
SRAM	2 Kbytes	**16 Kbytes**
EEPROM	1024 bytes	**4096 bytes**

Figure 4. Comparison between 328p (Uno) and 1284p.

3.2 3xDS18B20 temperature sensors

The reason for using these temperature sensors was the possibility of measuring a temperature between -55C and +125C, required temperatures since a GPU could reach more than 100C, an already critical operating temperature (Figure 5). Another very important aspect was the ability to communicate with multiple sensors simultaneously on the same data bus with the unique 64-bit addresses stored in their own ROM, thus occupying fewer pins of the microcontroller, giving the possibility of a easier and simpler PCB design, with fewer components and saved space.

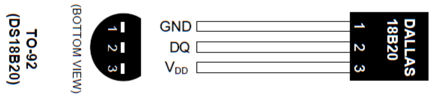


Figure 5. DS18B20 temperature sensor.

The main features of the DS18B20 [9]:

- One-Wire communication (single wire/bus communication);
- Single address for each sensor;
- Does not require external components except a 4.7kΩ resistor;
- Power supply between 3V and 5.5V;
- Measurement of temperature can be between -55C and + 125C;
- Accuracy of 0.5C from -10C to +85C;
- User defined resolution ranging from 9 to 12 bits;
- Converts 12-bit word temperature to 750ms;
- Alarm setting.

3.3 Sound sensor FC-04

The sound sensor FC-04 (Figures 6) recognizes the sound waves with a comparator between the microphone and a reference voltage (potentiometer) and gives a higher or lower voltage (0 or 1) in accordance with heard sounds.

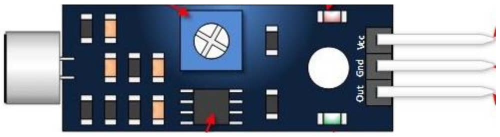


Figure 6. FC-04 sound sensor.

Technical specifications [10]:

- Power supply at 3.3V -5V;
- Output 0 or 1, High or Low;
- PCB 3.4cm x 1.6cm.

3.4 RTC I2C DS1307 Module

The module was chosen because it has the main functions of I2C communication, its own separate power supply, and it is necessary when we want to see when a certain temperature or state of the circuit has been detected (Figure 7).

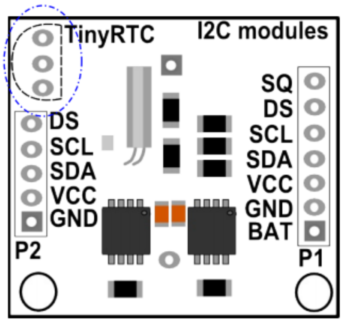


Figure 7. DS1307 module.

The main features of the module are [11]:

- I2C communication with its own address;
- Separate power supply 3V;
- Possibility to integrate a DS18B20;
- Remember h/m/s, D/M/Y;
- It can also be powered by the circuit.

3.5 UART communication for serial

This type of communication is used to interact with the PC and MCU programming. To do this, use a FTDI with a USB conversion circuit to RS232 and vice versa [12]. It also has the role (in the prototype circuit) of 5V supply of USB to the whole circuit (Figure 8).

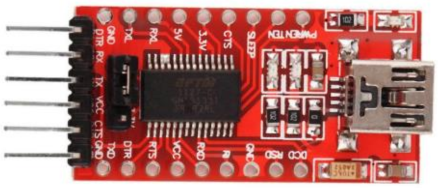


Figure 8. FTDI 232.

3.6 Other components used

1. Potentiometers 10k for adjusting the 3 fans, the 3 colors or other functions;
2. Buzzer for audible warning at 5V;
3. Quartz of 16Mhz together with 2 capacitors of 22pF;
4. RGB LED (or RGB strip with transistor TIP120) for visual warning, appearance and light effect for sound detection;
5. Various resistors: 230Ohms, 4.7k, 10k, 1M, etc. Resistors used for LEDs, to cancel voltages, voltage divider, Reset resistor;

6. Power LED, green light for power on;
7. 104uF condenser for filtering;
8. Switch Reset;
9. Connection and bridging cables;
10. 16x2 LCD LED with I2C module [13].

4. CIRCUIT

4.1 Prototype 1

Prototype 1 was the first working circuit built on a breadboard presented on ZAST 2018 at University of Valahia in Targoviste (Figure 9).

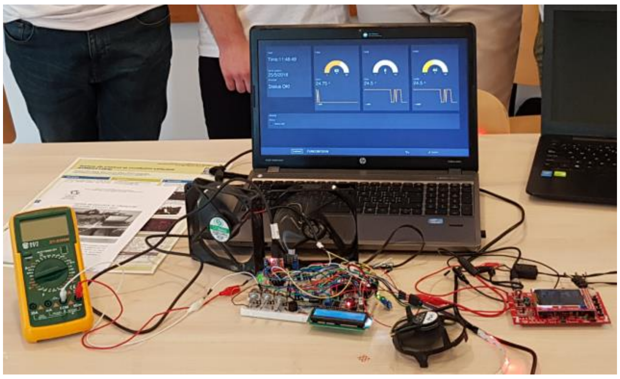


Figure 9. Prototype 1.

4.2 Prototype 2

Prototype 2 was the updated PCB test version of the circuit, presented at SCCSS-IEECC 2018 at University of Valahia in Targoviste (Figure 10).

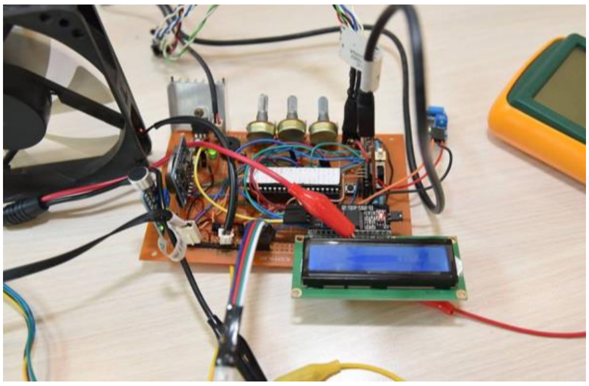


Figure 10. Prototype 2.

4.3 Proteus Prototype 3 PCB

Prototype 3 is a concept look for the circuit made in Proteus v8 (Figure 11).

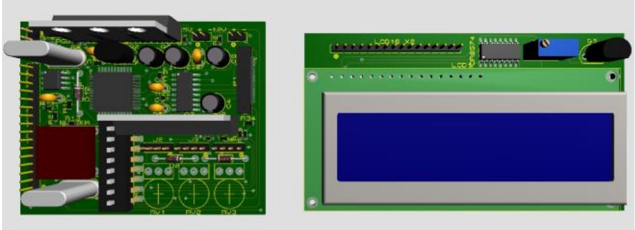


Figure 11. Proteus Prototype 3.

5. CONCLUSIONS AND FINISHED CIRCUIT

The proposed system is a low cost (less than 100 ron) that offers the possibility to manually adjust fan speed, in which case the user is directly responsible for system behavior, the high temperature alarm and Website transmission are still active in this state [14].

The system can also take its own fan speed decisions after a minimum and maximum value set by the user, the alarm and Website transmission are also active in this state.

The total system size can be reduced by 25% -35% (value depending on the components used and PCB layers) compared to the size of an Arduino UNO.

In the music Lightshow or during color setting (RGB), the monitoring and control code, as well as the alarm and transmission code on the Website, are deactivated until the switch is acted on in its normal working position.

The need for transistors for fan control brings a small distortion to the PWM signal (Figure 12 and Figure 13), consisting of a ripple because of poor connection to the breadboard and transistor quality (partially solved on prototype 2).

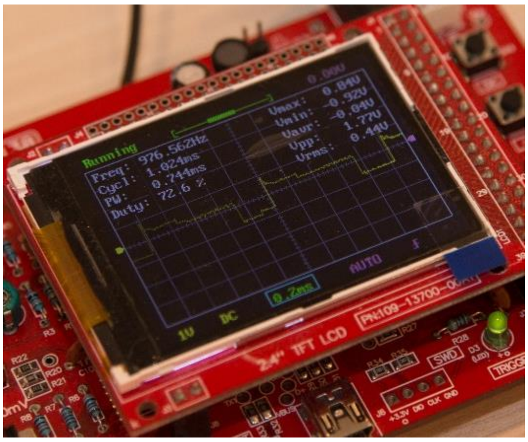


Figure 12. PWM ripple.

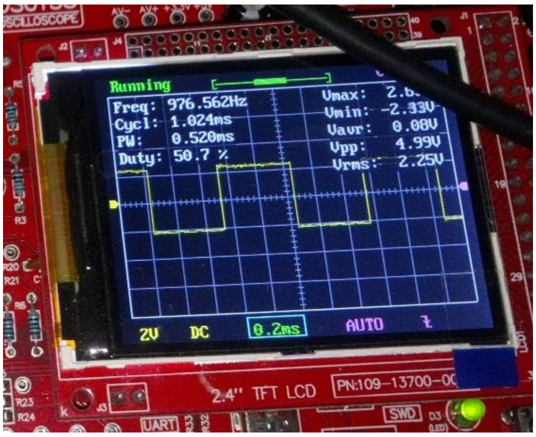
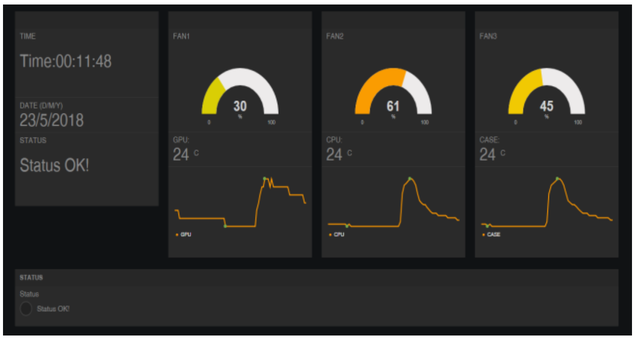


Figure 13. Normal PWM.

Display on the Website is also dependent of the computer speed to which the system is connected and the internet speed. In optimal conditions we have achieved a transmission speed of only 1 second, but under less favorable conditions the delay can be even over 9 seconds. The problem can be solved by a Wireless module.

Time and Date are displayed directly from the circuit, the speed control is represented by a "Gauge", Status is displayed once with the Ok message and the LED is extinguished, and in case of over temperature, the LED lights up and the high temperature alert message as well as the fan distributed to the sensor and the temperature is displayed both by message and graph (Figure 14).



TIME
Time:00:11:48
DATE (D/M/Y)
23/5/2018
STATUS
Status OK!

FAN1
GPU:
24 C

FAN2
CPU:
24 C

FAN3
CASE:
24 C

STATUS
Status
○ Status OK!

Figure 14. Website.

By controlling the PWM with transistors and certain frequencies (in this case 978Hz on 2 pins and 476Hz on pin 3) in the fan a high frequency noise is heard because the voltage does not come in continuous but interrupted by the PWM signal with a frequency of 987/476 Hz.

Under maximum and even minimum speed, there is a loss of voltage between 0.7V and 1.8V once due to the consumption of the whole circuit, the problematic connection and the transistors.

The minimum voltage for the 12V fan to work is somewhere at 4-5V (depending on the size of the fan), for which although the circuit gives a small signal, the fan may start from about 20% but the fan can be held at a low speed without letting it stop (below 3V).

Another problem was the minimum and maximum reading or writing value that could not be 0 or 100% due to the imperfect connection, these values being reached only if the components were mounted on a PCB (problem solved on prototype 2).

The circuit is also able to operate with any kind of voltage fans by simply changing the transistors.

The system can be used in a desktop and implemented in any industrial or private domain that has to do with cooling/heating/overheating/Smart Housing/Automotive etc [15-16].

6. ACKNOWLEDGMENTS

I want to thank the teachers that helped and encouraged me to do the project, especially s.l.dr.ing. Gabriel PREDUSCA, s.l.dr.ing. Ion CACIULA and conf.dr.ing. Henri-George COANDA.

7. REFERENCES

[1] Paleru V., Sisteme de achizitii si interfete. Note de curs, 2013.
[2] Di Paolo Emilio M., Data acquisition systems – from fundamentals to applied design. New York: Springer, 2013.
[3] Predusca G., Bazele sistemelor de achizitii de date. Note de curs, 2018.
[4] Satish Kumar Peddapelli, Pulse Width Modulation. Analysis and performance in multilevel inverters, De Gruyter Oldenbourg, 2016.
[5] Gibb A.M., New media art, design, and the Arduino microcontroller: a malleable tool, Thesis, Pratt Institute, 2010.
[6] Atmega 1284p - https://github.com/maniacbug/mighty-1284p.
[7] Arduino CC - https://www.Arduino.cc.
[8] Arduino Uno Rev3 - https://store.Arduino.cc/Arduino-uno-rev3.
[9] Arduino modules – DS18B20 - http://arduino-info.wikispaces.com/Brick-Temperature-DS18B20#multibus.
[10] Sound sensor module – FC-04 - https://www.optimusdigital.ro/ro/senzori-altele/108-modul-senzor-sunet.html.
[11] I2C RTC – DS1307 - https://datasheets.maximintegrated.com/en/ds/DS1307.pdf.
[12] FTDI 232 - https://stak.com/USB_to_TTL_Serial_Adapter__FTDI_FT232RL_chipset__3.3V_and_5V_compatible.
[13] 16x2 LCD LED with I2C module – Arduino LiquidCrystalDisplay, https://www.Arduino.cc/en/Tutorial/LiquidCrystalDisplay.
[14] https://dweet.io/
[15] Tareq El-Hasa, et.al., Arduino and Labview based control for efficient drive of cooling fan system, Research Journal of Applied Sciences, Engineering and Technology 13(10), 2016, pp.771-780.
[16] Suraj Kaushik, et.al., Automatic fan speed control using temperature and humidity sensor and Arduino, International Journal of Advanced Research, Ideas and Innovations in Technology, Volume 4, Issue 2, 2018, pp.453-457.