Research and Design of Automobile's Automatic Wipers Based on FPGA

Yezhi Yin[1, a], Ke Liu[2, b], Zhijun Li[3, c]

[1, 2, 3] School of Automation, Wuhan University of Technology, Wuhan 430070, P. R. China

[a, b, c] yinyezhi@whut.edu.cn

Keywords: FPGA, PWM, PID, Rain sensor

Abstract: The Field Programmable Gate Array(FPGA) can be used to generate PWM waves to control DC motor's working, therefore a system of automobile's automatic wipers which could convert wipers' speed autonomously were designed base on it in this paper. The core module of the system was FPGA which generated PWM waves in reference to the command-signal received, and the PWM waves drove the DC motor that made wipers work, the FPGA module also had a PID algorithm to ensure that DC motor could work stably and accurately. A novel rain sensor which was termed "modified RHD-22 rain sensor" has been put forward in this paper to give out this command-signal according to the rain it detected. Result of the simulation showed that the system could achieve almost all the functions needed.

Introduction

With the developing of technology, automobile has become a part of ordinary families. But hidden danger came along with this, too. There is so much information---such as situations, signal boards and sudden events---need to be disposed by drivers that accidents may happen when the driver has to control wipers in addition to above things in a raining day. Thus an automatic wiper which can change its speed independently is being required urgently.

The core of the automatic wiper system is DC motor, to control the DC motor's speed the most widely-used technology is PWM (Pulse Width Modulation). Former PWM was wholly achieved by hardware circuit, neither the circuit components were simple nor the PWM wave was accurate; Motors which have a MCU (Microcontroller Unit) inside had been used to generate PWM wave， but it couldn't be equipped with an algorithm due to its hardware structure; Although the developing of digital circuit gives us a flexible and fully functioning way to generate PWM wave, the achievement of its functions depends on peripheral circuits so the anti-interference ability is worse than FPGA. Therefore, taking cost and flexibility into account, FPGA has great advantages in DC motor's controlling [1, 2]. As an electronic application engineer in 21st century, abilities to design an electronic system based on FPGA is a fraction of the skills essential, FPGA has a set of integral language, designers can design, simulate, test and verify the circuit on the computer, then realize it at a special chip. This makes the consumption much lower, boosts the reliability and diminishes the development cycle, besides it can be upgraded and maintained if necessary, so the cost of a FPGA-based circuit is quite cheap [3].

General Design of the System

The system's main modules are as follows: firstly, a module that can perceive the rain was needed; it can give out a proportional signal while it perceiving the current rainfall, so that following modules can operate depends on this signal. Then, a module that can process data and signals received was necessary, in addition, this module can also control the DC motor's speed directly, at the same time, we used a PID algorithm to keep the DC motor running stably, the PID algorithm and the disposal of the signal which is returned by tachogenerator were achieved in this modules, too. Finally, we added some modules about the motors, to integrate the system. The system's frame was shown in Fig. 1.

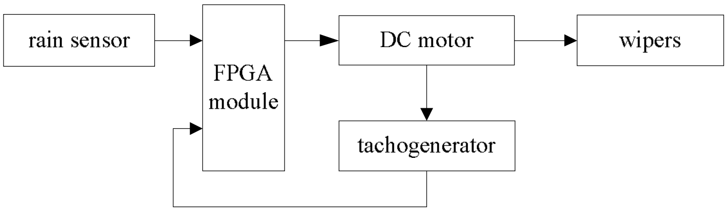

Figure 1. frame of the automatic wiper's system

The automatic wiper system was triggered by the rain sensor. When there were raindrops on the automobile, the rain senor generated a signal whose frequency would be raised with the increase of the rainfall in a certain proportion. After FPGA module's disposal, the signal was modulated by PWM to control the motor's speed as we needed. When the DC motor alters its orientation or how to keep the DC motor's speed accurate was respectively judged by signals came from the limit-switch and the tachogenerator after FPGA module received and analyzed them. In the end, the wipers were fully dominated by the DC motor. The whole process diagram was shown in Fig. 2.

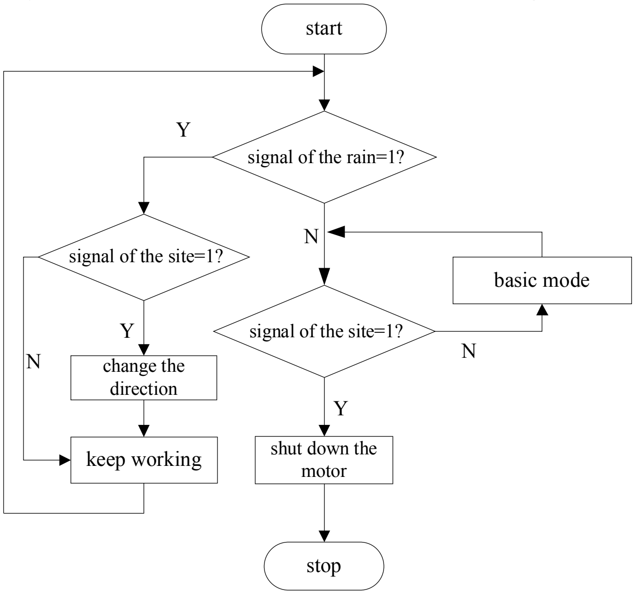

Figure 2. diagram of the system's process

In this diagram, "signal of the site" means signals emitted by the limit-switch. Every time the wipers touch this machine, it will generate a signal to "tell" the system "wipers in groove". Signals' reception, logic judgments and motor's control are completed in FPGA module at all.

Modified RHD-22 Rain Sensor

The "modified RHD-22 rain sensor" is a rain sensor especial for automobile that authors reformed from a RHD-22 rain sensor (Fig. 3), it solved a ubiquitous deviation in infrared rain sensors which caused by their indirect sensation [4].

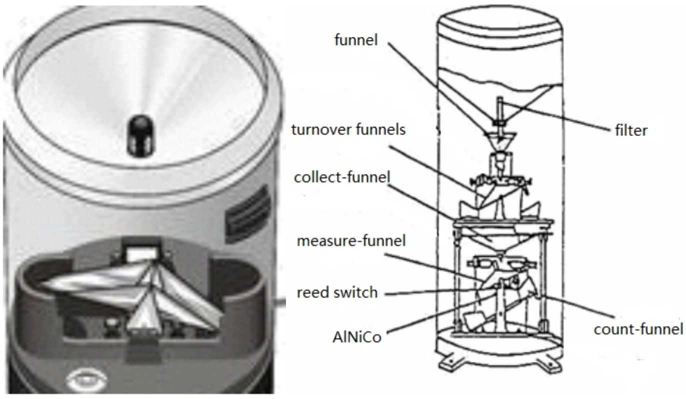

Figure 3. modified RHD-22 rain sensor

Because of the low accurate requirement, modified RHD-22 rain sensor only has a funnel, a filter, two turnover funnels, a reed switch, a piece of AlNiCo and so on, its structure is simple. Rain collector (superseded by wipers groove on the automobile) collects the rain to the funnel and the funnel delivers it to one of the turnover funnels through the filter, another turnover funnel will replaces this one's work by reversal if it has collected rains enough. They work in turn and every time they hand over their works, the little piece of AlNiCo on each of them will make the reed switch connect once and generate a switching signal [5].

This switching signal would be "translated" into a speed data in this system—it means collecting a group of data which represent the speed needed to keep windows clear on the premise of the given signal via experiment, and fitting them into a linear function by Least Square Method—then the DC voltage that DC motor needed to keep the speed would be calculated relied on DC motor's rated parameters, so that we can ascertain the duty ratio of PWM. All the data emerged above was stored in the registers of FPGA, provides to following modules.

FPGA Module

FPGA module's tasks mainly include two parts: using a PID algorithm to control output signal depends on the feedback signal and generating a PWM wave combine the output signal with the modified RHD-22 rain sensor's signal (Fig. 4).The codes of these two parts' hardware design were written on SOPC Builder and created a Nios ii system, the codes of these two parts' software design were written and debugged on Nios ii IDE. SOPC (System-on-a-Programmable-Chip) based on a large scale of FPGA, there are many I/O ports, memories and CPU integrated on it so that it's flexible on design, changeable on LE's gross and able to be upgraded. The writer has adopted a type of Cyclone II chips termed "EP2C5T144C8-FPGA" which was produced by Altera Company, it's equipped with the characters of high-speed, low consumption, high-frequency, flexible, cheap and so on [6]. It's designed with look-up-table structure based on SRAM which provides with 200 thousands Les.

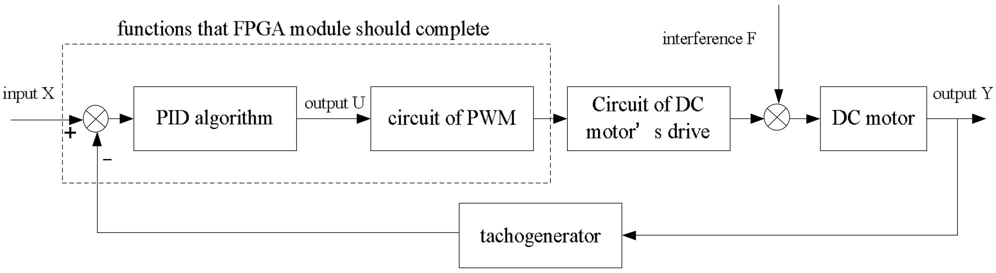

Figure 4. structure chart of the system

1) Selection of the algorithm

It's quite important for an operational system to keep stable, thus a tachogenerator has been used in this automatic wiper system to make it a closed loop, and added an algorithm in it as well, to ensure the stability of the system. Proportional Integration (PI) algorithm has betrayed the system's requirement that it must recovers the window's clear quickly because it lags much in response to the interference and needs too much time to complete the work; Due to the existence of the steady-state errors, Proportion Differentiation (PD) algorithm can't give the exact output; thus we combined the two algorithms' advantages and chose the Proportion Integration Differentiation (PID) algorithm which could strengthen the system's stability, restrain the maximum error during regulation process and remove steady-state errors while boost the system's speed of response [7].

The operation law of the PID regulator is

$$u_{(t)}\Big|_{t=k} \approx u_{(k)} = \frac{1}{p}\left\{e_{(k)} + \frac{T}{T_i}\sum_{i=0}^{k} e_{(i)} + \frac{T_d}{T}\left[e_{(k)} - e_{(k-1)}\right]\right\}. \tag{1}$$

In the equation (1), $u_{(k)}$ means the output of the PID at the moment of K, $e_{(k)}$ means the input error at the moment of K, T is the sampling period of the digital circuit and p, T_i, T_d represent for the proportional coefficient, the integral time constant and the differential time constant which we want to know. The formula has used the whole accumulation before the K moment (that is, $\sum_{i=0}^{k} e_{(i)}$), so it's easy to emerge great errors. Then we had to modify the formula as:

$$\Delta u_{(k)} = u_{(k)} - u_{(k-1)} = \frac{1}{p}\{[e_{(k)} - e_{(k-1)}] + \frac{T}{T_i}e_{(k)} + \frac{T_d}{T}[e_{(k)} - 2e_{(k-1)} + e_{(k-2)})]\}. \tag{2}$$

$\Delta u_{(k)}$ is termed incremental algorithm and only three moments' errors are needed in this equation (2). A group of data about ($e_{(k)}, u_{(k)}$) was achieved from emulation experiment and built a mathematical model on the computer by which a group of suitable values of p,T_i ,$T_d$could be calculated. From the formula we can see that input errors $e_{(k)}, e_{(k-1)}, e_{(k-2)}$ is relevant to $x_{(k)}$-$y_{(k)}$, and the output $\Delta u_{(k)}$consists of this three errors, so that codes of this part's hardware was wrote by Verilog HDL language as [8,9] (portion of the codes):

```
module wucha (clk,X,Y,e0,e1,e2);
    input clk;
    input [7:0] X,Y;
    output [7:0] e0,e1,e2;
    reg [7:0] e0,e1,e2;
    assign e2=X-Y;
    always@ (posedgeclk)
        begin
                e1<=e2;
        end
    always@ (posedgeclk)
        begin
                e0<=e1;
        end
    end module
```

The emulation of the output on Quartus II when it was given a certain single input was shown in Fig.5.

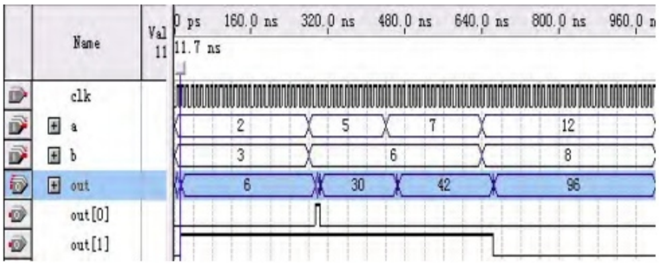

Figure 5. output of a single input

Adding the three outputs then we got the output $\Delta u_{(k)}$of the incremental PID algorithm.

2) Generation of the PWM

The basic principle to generate PWM wave is that narrow waves which have the same impulses but are different in their shapes can result in equal effect when they are applied to inertial links. Impulse means area and equal effect represent for same waveform, thus the DC motor's speed can be controlled by managing the on-off time of its input.

PWM module's hardware mainly completed three functions: the whole task's logic, producing a standard cycle and giving a proper output after comparing all the information. The whole task's logic operated synchronously via a simple clock, a counter had been used to provide a certain cycle and a duty ratio to PWM, and these data was set by a microcontroller. All the data that was produced by the functions above was stored in different registers and several interfaces, logical controllers as well, were provided to read or write these registers. Besides, the microcontroller could shut down the PWM module's output by operating the register's inhibit bit [10]. The whole logical task of PWM module was shown in Fig. 6.

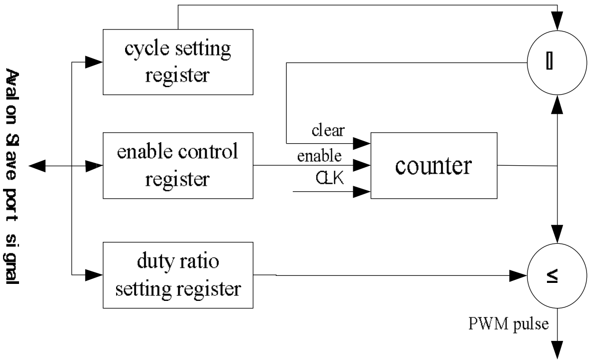

Figure 6. structure of PWM module's logic

The circuit compared the current value in the counter with the value which had been set in the register of duty ratio before to decide the output level. If the former is not bigger than the latter, it gives a low level; otherwise it gives a high level. The register that was used to set a standard cycle provided a clock signal which was controllable at its reference. A PWM wave's cycle was over when the value in the counter ran up to the standard. Then the value should be swept. Enable register controlled the clock signal's validity to decide whether the PWM module gave an output or not. Base on the depictions above, we could write the codes of the PWM module (portion of it) by Verilog HDL language as:

```
assign pwm_enable=control_reg;
always @ (posedgeclk or negedgereset_n)
   begin
      if (reset_n==1'b0)
         PWM_counter=0;
      else
         begin
            if (pwm_enable)
               begin
                  if (PWM_counter>=clock_divide_reg)
                     PWM_counter<=0;
                  else
                     PWM_counter<= PWM_counter+1;
               end
         else
               PWM_counter<=0;
      end
end
always @ (posedgeclk or negedgereset_n)
begin
   if (reset_n==1'b0)
      PWM_out<=1'b0;
   else
      begin
         if (pwm_enable)
            begin
               if (PWM_counter<=duty_cycle_reg)
                  PWM_out<=1'b1;
               else
                  PWM_out<=1'b0;
            end
         else
            PWM_out<=1'b0;
```

```
        end
    end
```

After we have finished the entire hardware's program (PID algorithm & PWM module), it could generate a Nios II system in SOPC Builder and was compiled in Quartus II [3]. Finally, we designed the software of the entire system (mainly about PID algorithm & PWM module) via C/C++ language (Concrete program wasn't shown here due to its diversity and ubiquity), and downloaded them to the chip after we compiled it in Quartus II.

Driving Circuit of Motor

The driving circuit of motor was shown in Fig. 7. PWM wave was input the DC motor through a H-bridge inverter circuit which was constituted by four MOSFETs, they were divided into two groups by whether they were on the diagonal and each group controlled a direction of the current (positive or negative) and then they controlled the direction of the DC motor. Every MOSFET had been equipped with a capacitance to moderate the voltage on it [11].

MOSFETs were controlled by integrated chip IR2103 which is high-speed at on-off and stable at working. Each chip can control two MOSFETs and is just to control the MOSFETs on the diagonal. The principle of the chip to control MOSFETs was shown in Fig. 8, SIGN1 (2) is a signal which was generated by a switching element at the ultimate position of the wiper's route when this element was triggered by the wiper. Only one switching element can be triggered at the same time.

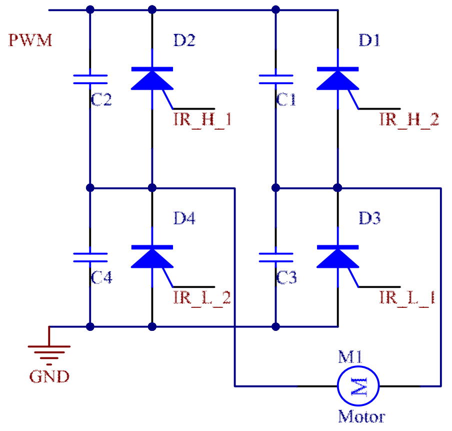

Figure 7. the driving circuit of DC motor

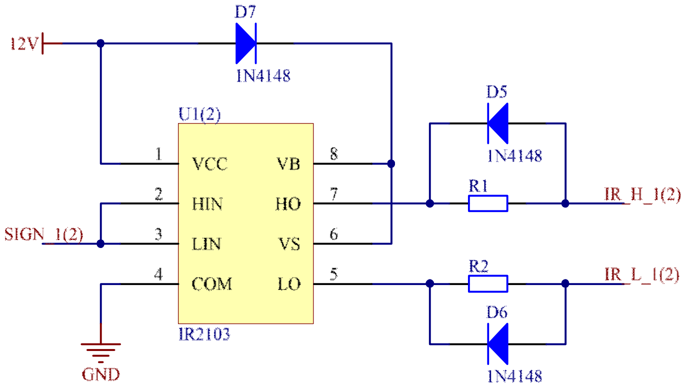

Figure 8. MOSFET's driving circuit

Conclusions

The authors came up with a novel rain sensor for automobile on the basis of analyzing the prevalent infrared rain sensor. The rain sensor has improved the sensitivity to the rain. Given the wipers on the automobile at present are dull at operation, can't convert its speed in line with the rainfall and have a large consumption on the resource, so the system had been adopted a continuously variable speed function that was based on PWM. This method can keep the windows clear while let the wipers working as economical as possible. More than this, the PWM was achieved by FPGA and benefited from its flexibility. The system had access to be modified in detail or be upgraded its hardware for a new idea in the future because of the FPGA's application, so it can be used by customers for a long time. The authors had also discussed this project's feasibility and practicability in reality from several aspects like the theory, the price, the functions etc. At the end of the design, the system was emulated in Quartus II and the result show that it can almost achieved all the functions we expected, this also confirmed this project's perspective in real life.

Acknowledgement

In this paper, the research was supported by Nature Science and Technology Major Project of China (2011ZX4002-21).

References

[1] Monmasson E，Idkhajine L，Naouar M.W. FPGA-based controllers [J].Industrial Electronics Magazine，IEEE，2011, 5 (1): 14~26.

[2] Milivojevic N，Krishnamurthy M，Gurkaynak Y etc. Stability analysis of FPGA-Based control of brushless DC motors and generators using digital PWM technique [J].Industrial Electronics Magazine，IEEE，2012, 59 (1): 343~351.

[3] Wang Gang, Zhang Lian. The SOPC FPGA-based embedded system design and typical examples [M].Beijing: Publishing House of Electronics Industry, January, 2009. (In Chinese).

[4] Li Wangpeng, Li Zhijun, Li Huijie etc. Design of novel infrared rain sensor [J].Journal of Wuhan University of Technology (Information & Management Engineering), 2013, 35 (1): 30~35. (In Chinese).

[5] Baidu library. Rainfall tester [EB/OL]. http://wemku.baidu.com/view/, September, 2010. (In Chinese).

[6] Liu Xuewei, Li zhijun, Fan Hurong etc. Design on control system of DC motor's speed based on FPGA [J]. Journal of Wuhan University of Technology (Information & Management Engineering), 2013, 35 (1): 30~35. (In Chinese).

[7] Wang Zaiying, Liu Huaixia etc. Process control system and meter [M].Beijing: China Machine Press, November, 2009. (In Chinese).

[8] XU Zhong-ren, JIANG Li, MU Ke etc. Digital PID controller based on FPGA [J]. Journal of Liaoning Shihua University, 2013, 32 (3): 74~79. (In Chinese).

[9] Peng Canqiong, Liu Gang, Dong Hu etc. Design of FPGA-Based incremental PID intelligent humidity controller [J].Science and Technology Innovation Herald, 2014, (3): 41~42. (In Chinese).

[10] Yang Jun, Cai Guanghui, Huang Qian etc. Design and practice of digital system based on FPGA [M].Beijing: Publishing House of Electronics Industry, January, 2014. (In Chinese).

[11] He Zikuan. Several kinds of driving circuit about mini-DC motor in automatic control [J]. Practical Electronics, 2014, 8 (2): 31~32. (In Chinese).