VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Daniil Bushuev

PID Control of Motor Using FPGA

Technology and Communication
2022

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Daniil Bushuev
Title	PID Control of Motor Using FPGA
Year	2022
Language	English
Pages	32 + 1 Appendix
Name of Supervisor	Santiago Chavez Vega

Motors are very widely used in our modern life. Every vehicle and vast majority of house appliances that are related to motion of any kind, contains one. One of the most popular types of motor is a BLDC which stands for Brushless Direct Current Motor. The speed of the motor would have been controlled by a PID controller implemented on a FPGA kit using VHDL.

FPGA programming and VHDL are not the most common solutions for previously stated goals, but it's difference from usual programming languages and unique features which seemed an interesting challenge, as well as possibility of broadening horizons it the field of study were reasons enough to use them as a base of the project. The goal of the thesis was to learn this unfamiliar field as well as study the principles of BLDC motor control.

During the work on the project, it turned out that building a driver for standard BLDC motor was challenging enough for a separate project such as thesis on its own. The decision was made to use a generic computer fan with PWM control that has a BLDC motor inside it, and which could be controlled by a digital signal from GPIO pin on an FPGA kit.

The results shown that FPGA units are capable of controlling the speed of the motor in a smooth and accurate manner, however it remained an obstacle to read the speed of the fan, so the system itself remained in an open loop state.

Keywords BLDC, motor, FPGA, VHDL, PID, PWM.

CONTENTS

LIST OF ABBREVIATIONS

BLDC	Brushless Direct Current
FPGA	Field Programmable Gate Array
VHDL	Very High-Speed Integrated Circuit Hardware Description Language
PID	Proportional Integral Derivative
PWM	Pulse-Width Modulation
IDE	Integrated Development Environment
GPIO	General Purpose Input Output
RPM	Rotations Per Minute
UART	Universal Asynchronous Receiver Transmitter
GUI	Graphical User Interface

LIST OF FIGURES AND TABLES

LIST OF APPENDICES

1 INTRODUCTION

BLDC motors have a very wide range of applications, from house appliances to manufacturing tools and vehicles. Ability to control the speed of a motor fast and smoothly is very important in most cases and sometimes crucial, for safety as well as time consumption.

FPGA (Field Programming Gate Array) is an integrated circuit that can be programmed by a user for a specific use after it has been manufactured /6/. They are flexible in utilisation and handling which “makes them very different from other types of microcontrollers or Central Processing Units (CPUs), whose configuration is set and sealed by a manufacturer and cannot be modified” /6/. It is not as widely spread, but its unique features and rarity is what made it stand apart from other solutions and was the reason it was chosen for this project.

Combining the two together, the idea of this thesis was that the FPGA kit, in this project the Altera DE2 kit, would be connected to a BLDC motor, and it would also have a PID controlled implemented on it with VHDL, which is a description language used to describe hardware and is utilized in electronic design automation to express mixed-signal and digital systems, such as ICs and FPGA /7/. That would allow the FPGA kit to control motor accurately and without unnecessary delay of acceleration. The kit would also be connected to a display that would be able to show the data of processes ongoing inside the FPGA.

2 THEORY

This part of the thesis covers the theory basis of the project. The part is divided into three subsections which are the most important for understanding how the system works, which are PID controller, PWM control and UART communication.

2.1 PID Controller

The Proportional Integral Derivative (PID) controller is a control system that can automatically detect error and change the processed values accordingly. It is used to control wide variety of parameters from temperature, pressure, water level to motor speed. In its essence the PID controller is closed loop system, which means that it gets the feedback each time the process is done and if the result is different from the setpoint it starts again. The difference between the setpoint and feedback value is called an error. The PID controller calculates the error by calculating the difference between the actual value and the desired value and then sets the deciding parameters accordingly. This error is continuously being calculated until the process stops.

The proportional part of the PID controller is used to calculate the error between the set value and the actual value and is responsible for the corrective response. The integral part is applied to take all the previous error values and then integrate them to find out the Integral term. When there is no more error in the system the integral part stops increasing. The derivative part is used to calculate the future error values based on the present values so far. The control of the system can be increased if the system has a dramatic change, which is also a responsibility of thr derivative part. A combination of all three of these parts makes the name Proportional Integral Derivative (PID) Controller.

The mathematical formula of the PID Controller can be presented as

$$ut = Kp\ et\ +\ Ki0tet‘dt‘\ +\ Kdde(t)/dt \text{ (1)}$$

In which:

Kp is the coefficient of Proportional term.

Ki is the coefficient of integral term.

Kd is the coefficient of derivative term.

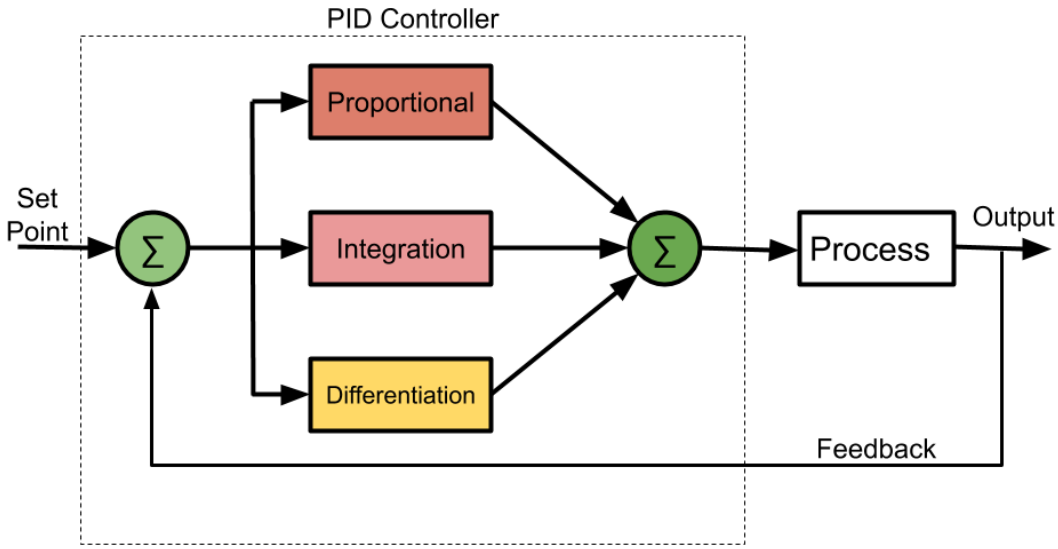


Figure 11. Block diagram of PID controller

In the beginning the input goes to the P, I and D at the same time and their output is summed together. This PID output is applied to the process which configures all the necessary parameters. The set of parameters is then sent back to the system as feedback. This feedback and the input from the beginning are then summed to calculate the error which then again is fed to the PID, and the loop continues until the error goes to zero or the system reaches the steady state, which means that

the output value of the whole system is not going to change, but it may nevertheless differ from the original input. The difference between the original input and a steady state value is called a steady state error.

PID Controllers usually contains all three parts (i.e. P, I and D) however sometimes processes utilize only two or even one of these units which results in PI, PD, P or I Controller. This can be achieved by setting the unused parts to zero which renders those parts non-existent and their effect on the system to zero.

2.1.1 Proportional Term

The output value that is produced by the proportional term is proportional to the error value at the given cycle. The error is multiplied by the coefficient "Kp" which allows the tuning of the values. Kp has another name which is proportional gain constant.

$$Pout = Kp * e(t) \quad (2)$$

If the value of the proportional gain constant is high, the system reaction for any given error will be more significant. If the Kp constant is too high for the system, it may become unstable. If the proportional gain is too small, then the output results will be further from the setpoint, meaning that the system becomes unresponsive, and it may go through disturbance or lagging. Thus, the correct tuning of proportional terms is crucial for creating a sensitive and responsive yet reliable system. The proportional control term is also responsible for the mechanism of the steady state error which is inversely proportional to the proportional gain.

2.1.2 Integral term

The integral term value depends on the magnitude of an error and its duration. The integral term sums all different instants of the error value in time. After that the result is multiplied by the integral gain coefficient and added to the system output.

$$Iout = Ki * \int_0^t e(t)'dt \quad (3)$$

The integral term increases the rate of change of the value in achieving the desired values and reduces the error caused by the proportional term. However, because the integral term calculates the sum of values from the past, its calculations may cause the system to overshoot the set values, which means that the actual value of the system gets higher than the set value.

2.1.3 Derivative Term

The derivative term is calculating the derivative of the error to estimate the slope of an error in time domain and then multiplies this derivative with derivative gain coefficient Kd.

$$Dout = Kd * \frac{de(t)}{dt} \quad (4)$$

The derivative term predicts system values and behavior in general to speed up the system and makes it more stable. Derivative terms are quite often left out from PID Controllers due to their usually insignificant impact on the system stability in practical application.

2.1.4 Stability and Characteristics

The three parameters of a PID controller are gain, integral term and derivative term.

If the parameters are chosen negligently, the PID controller can be insecure meaning that the output of the system may not be as precise or accurate as it could be

with other values. Instability could be caused by significant unnecessary gain, especially when the delay of the system is quite high, in other words if the system lags, gain amplifies the instability probability.

Instability can be observed in the Laplace domain from the following formula /1/:

$$Hs = Ks * G(s)1 + Ks * G(s) \quad (5)$$

Where:

- Ks = PID Transfer Function
- Gs = Plant Transfer Function

The system stability can be measured by the result of the product of K(s) and G(s). There are these conditions, which are:

- If Ks*Gs= <1, this means the system is stable
- If Ks*Gs= -1, this means the system is unstable

The optimal behavior on a process or set point change varies depending on the application.

The two important characteristics of the system in time domain are Rise Time and Settling Time. Rise time is considered to be the time between the points where the system reaches 10% and 90% of the set value /2/. Settling time is the time when the system reaches and stays within the small error from the setpoint.

The Figure 2 explains the relation between the coefficient values and rise time, settling time and overshoot. The green line represents the situation, where all the coefficients are equal one. If we increase the integral gain to two, the system will have a less rise time but more fluctuation and overshoot, as black line shows. And the value of 0.5 of the Ki leads to bigger rise time, but no overshoots, as depicted

by red line. Each of the conditions could be useful, depending on the real-life application case.

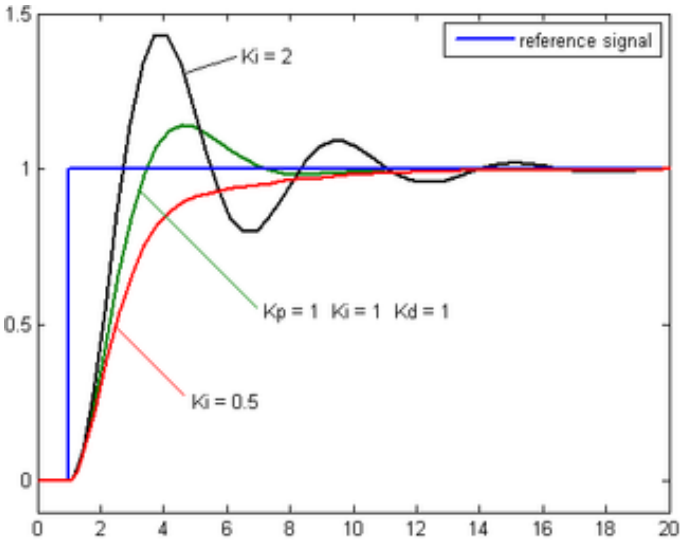


Figure 12. PID controller behavior comparison

2.2 Pulse width modulation

Pulse width modulation a digital output which can be used to control analog devices. The main characteristic of the PWM signal is duty cycle. Duty cycle is the relation between the time in the period where the signal is high (logical 1) to the whole period of the signal. If the period of a PWM signal is 4ms and during this period, the signal is high for 1ms then the duty cycle is equal 25%. If the time of high state is exactly equal the time of low state, then the duty cycle is 50%. Constant high state of the signal indicated a 100% duty cycle.

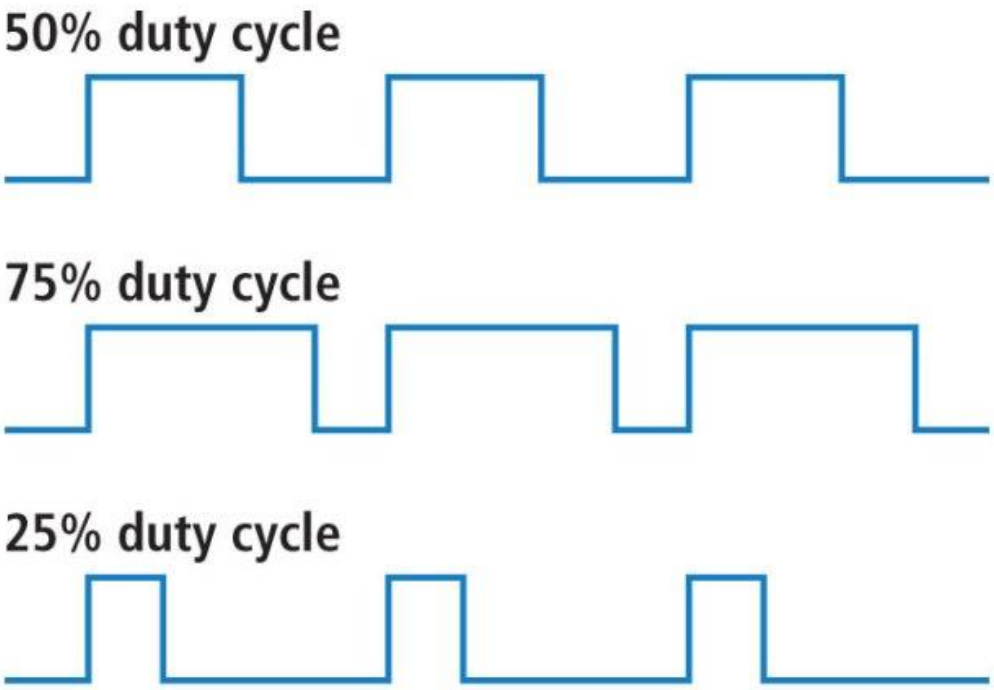


Figure 13. PWM signals with different duty cycle

The duty cycle affects the analog signal in a directly proportional manner. If the PWM controls the output voltage and the maximum output voltage value is 5V, then the output at 50% duty cycle would around 2.5V.

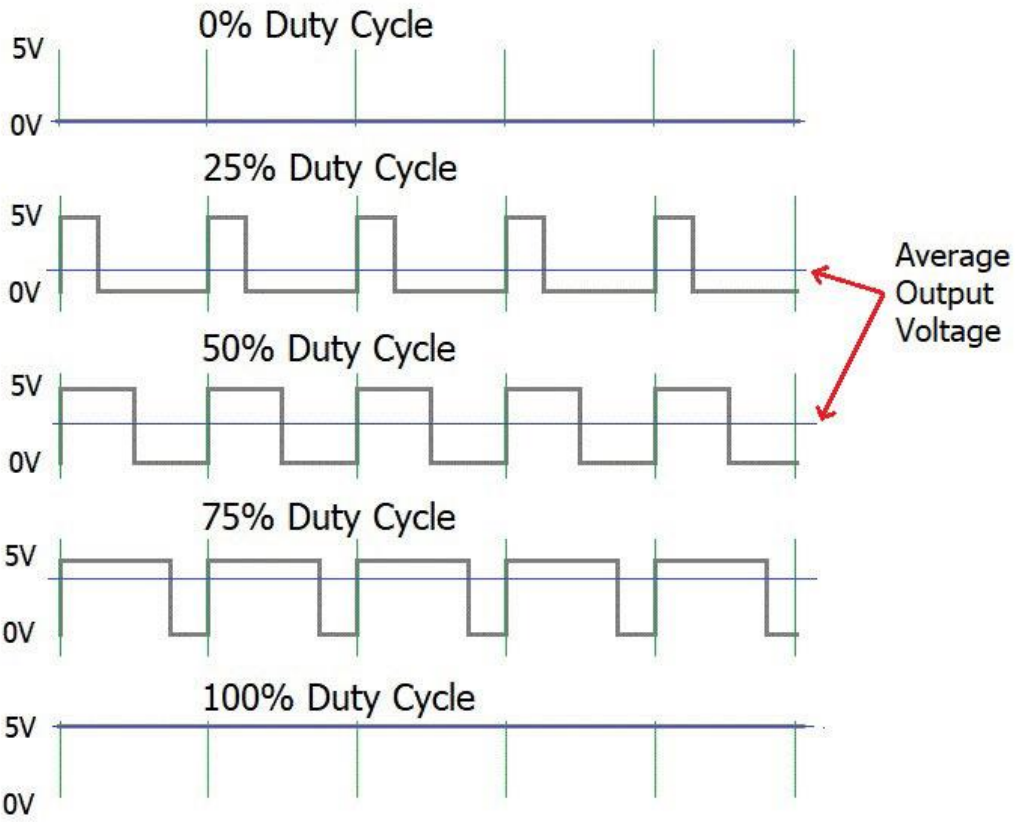


Figure 14. Correlation between the duty cycle and voltage level

In this project the PWM signal was used to control the speed of a fan. According to the datasheet provided by the manufacturer, at the 100% duty cycle the fan is supposed to rotate at 12000 RPM, at 20% the speed is 3000 rpm and at 0% the fan had to stay idle.

3 DEVELOPMENT

This section consists of the description of the development of the project including the hardware, software, and coding as well as the final testing and measurement.

3.1 Hardware

Two main hardware pieces were used in the project: an FPGA board that hosted the VHDL code implementation of PID controller, a PWM signal generator and other modules.

3.1.1 FPGA Kit

The FPGA board Altera DE2 (Figure 5) was chosen as the base for the thesis project.

Figure 15. Altera DE2 board

It was intended to be used as an educational board to study digital logic, hardware organisation and concept of FPGA. Moreover, this was the board that was used in the “FPGA and VHDL” course, so it was already familiar; in addition, this board, though being quite old, was known to have enough resources to handle a relatively light project.

3.1.2 Fan with BLDC Motor

Initially, the BLDC motor was supposed to be used, but, as it turned out, creating a driver for a bare BLDC motor, is a major project, probably even of a thesis scale on its own. So, the decision was made to use a regular fan with PWM control, which was simpler in implementation.

Figure 16. Sanyo Denki fan with PWM control

The fan that was eventually used was manufactured by Sanyo Denki (see Figure 6), has maximum speed of about 12000 RPM and has four wires: source, ground and PWM wire as well as tachometer wire that can be used to read the speed of the fan.

3.2 Software

Intel Quartus II Web Edition Design Software Version 13.0sp1 is the full name of the program that was used to upload the VHDL code onto the Altera DE2 board. This version of the software is not the latest one, in fact, it is quite old, but because of the age of the FPGA kit itself, they were compatible in all the ways that were important for the project.

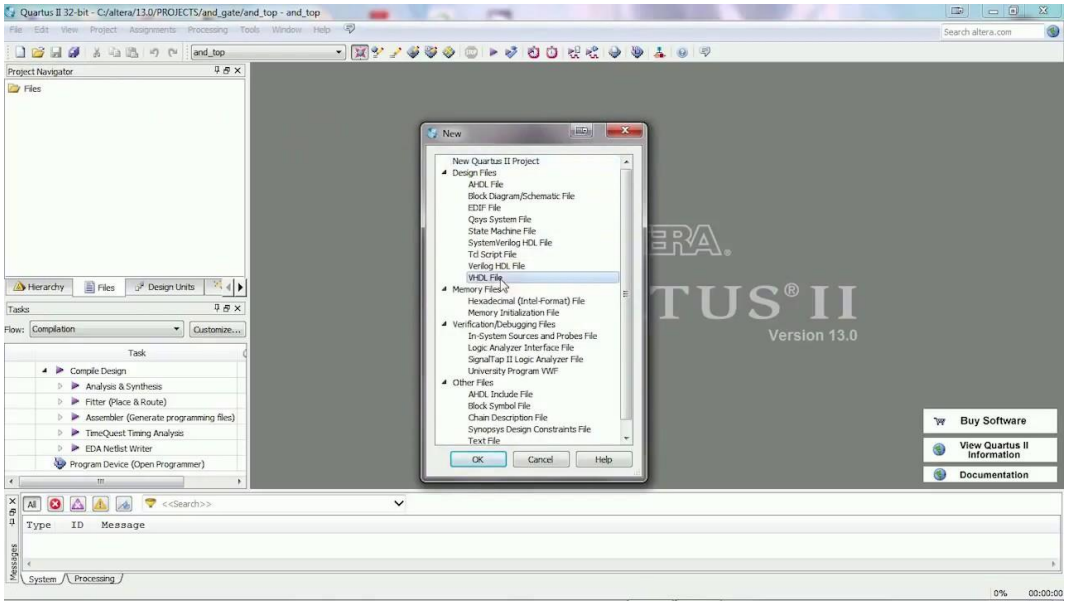


Figure 17. GUI of Quartus II

For a short period of time software named ModelSim was used to simulate certain modules before integrating them into the main project to test and debug errors without causing problems in other modules.

3.3 System Overview

This chapter concentrates on the overall system description, its functionality work-flow and circuitry.

3.3.1 Function Block Diagram

In this project, the FPGA kit has two connections to the fan; the first is the PWM signal and second is the tachometer wire, each connected to GPIO pins on the board. The Altera board has three modules on it implemented in VHDL: PID, PWN and UART. The PWM signal directly controls the speed of the fan and the PID controller was intended to change the duty cycle of the PWM. To generate a correct system response, the tachometer wire serves the purpose of the feedback value.

The UART module makes it possible to communicate with the PC via a terminal, to send information about processes happening inside the board.

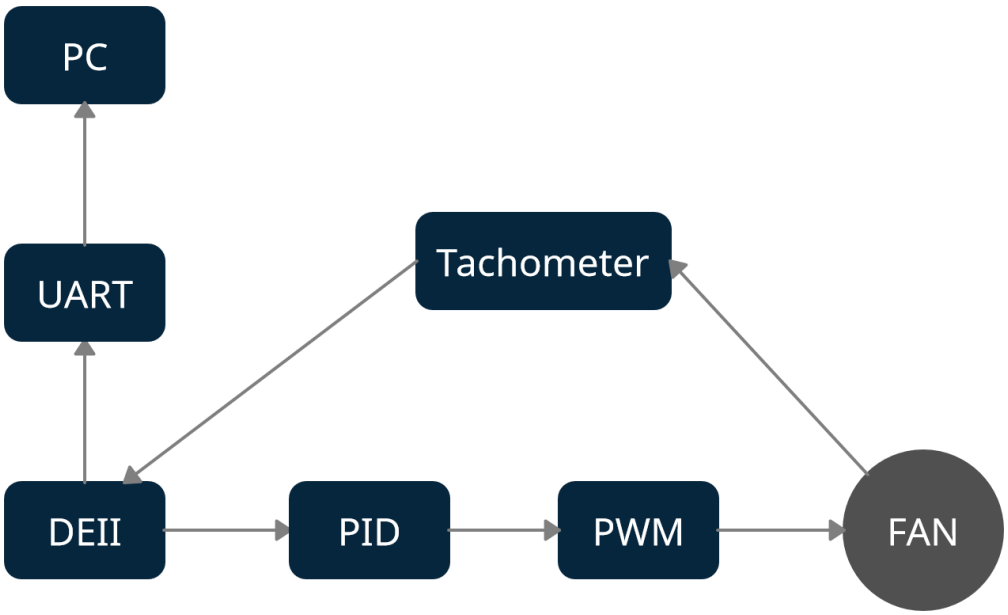


Figure 18. Functional block diagram

3.3.2 Circuit Diagram

The wire connection in practice is most simple. The fan is connected to the voltage source and both the FPGA kit and the fan are connected to the common ground and the PWM wire is connected directly to the GPIO pin that is set into the output mode. The tachometer wire however requires a pull up resistor, according to the manufacturer's datasheet. The value of the resistor was not known in the beginning, but in the same datasheet it is mentioned that the current on the tachometer wire had to be 5mA at most, and from that value it was calculated that the resistor must be at least 2.6kOhm, but the closest of the greater values that Technobothnia had at disposal was 2.7kOhm which worked just fine.

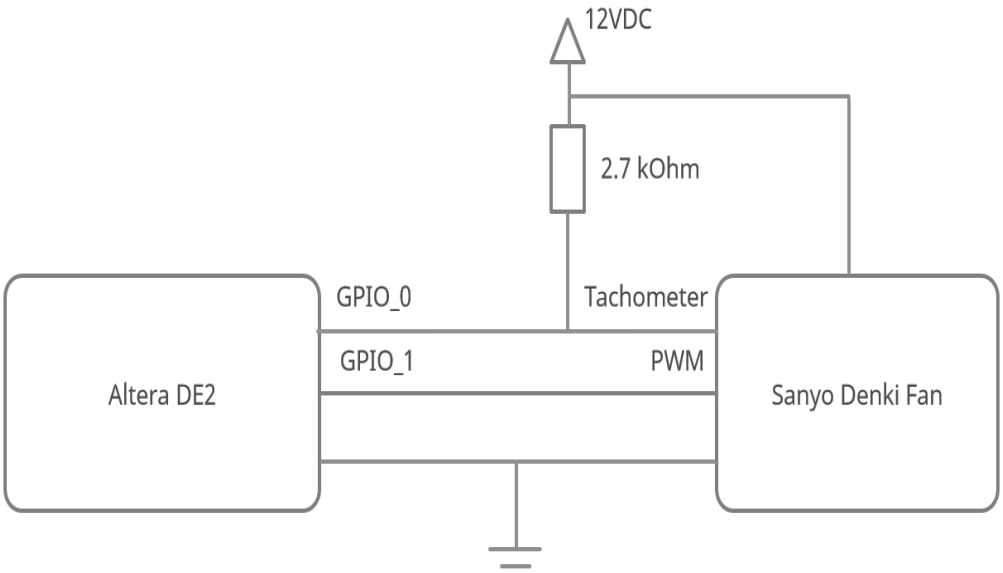


Figure 19. Circuit diagram

During the development process, the oscilloscope was used to monitor the state of the GPIO pin as well as the signals on the tachometer and PWM wires. The device was also used during the testing and measurement phase.

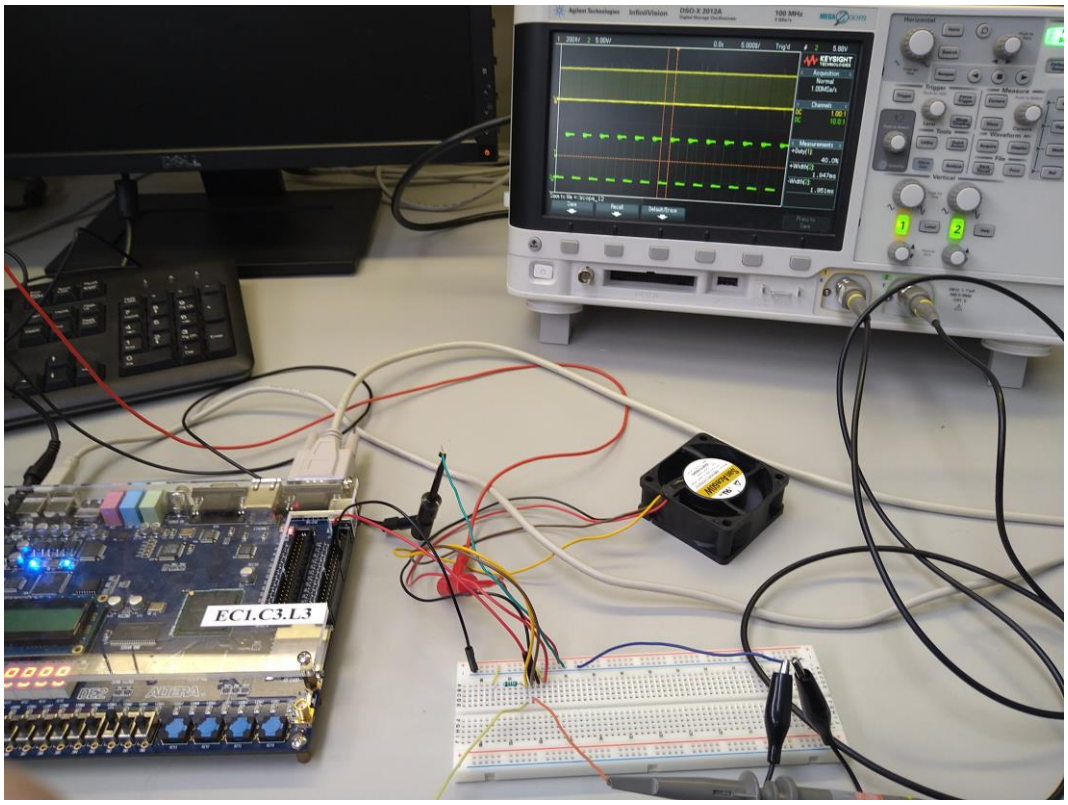


Figure 20. Assembled system with connected oscilloscope

3.4 VHDL code

The source code of the project consists of six VHDL files:

- counter.vhdl
- pid.vhdl
- pwm.vhdl
- pwm_led.vhdl
- reset.vhdl
- tx.vhdl

As the name of the file implies, counter.vhdl implements the code for measuring time and creating different time frames, using the high frequency clock, integrated in the board. The time slice that is calculated in this file is then used to tell the

PWM when to change the state of the signal, thus creating an accurate duty cycle, allowing the control of the fan.

The PID file is responsible for controlling the value of the duty cycle of the PWM signal. It uses the formulas described in the PID related theory part of the thesis, to calculate the system response in accordance with the feedback value that it gets through the tachometer wire, which is connected to the GPIO pin, set in the input mode.

The generation of the PWM signal happens in pwm.vhdl file with the help of the Counter. It puts the signal in the high state and defines the counter limit, which corresponds to a certain duty cycle. Then, when the counter limit is reached, the state changes to back low, and so the PWM signal can be sent onto the output GPIO pin.

The pwm_led.vhdl serves as a main function file, that includes and combines all other files in one program. Besides controlling, that correct values are parsed to correct modules, this file is also responsible for enabling necessary controls on the board. It enables switches, that are located on the board to let them control the duty cycle from 0% to 100% with a step of 10, as well as using a press button to send the values through the UART transmitter to the computer.

The tx.vhdl enables UART communication between the board and the PC terminal. It is responsible for creating the message with correct format, as well as setting the correct baud rate values, so that the message can be understood on the other end of the line. The project functionality did not require the receiver module, so the communication is not bidirectional.

3.5 Testing and Measurement

When modules were combined, compiled, and uploaded on the FPGA board, the testing of the system was performed. Figures 11-21 show the measurements done with the oscilloscope. Channel 1 (yellow line) represents the PWM signal, that is sent from the Altera DE2 to the fan and channel 2 (green line) is the tachometer output of the fan. The trigger is set on channel 2 because its frequency is much lower than channel 1 and if the trigger is set on PWM, the edges of the tachometer signal are outside of the scope and the device cannot measure it. On the right side of figures 11-21 there are two measurements, the duty cycle of PWM and the on-time of the speed sensor of the fan, which allowed to calculate the speed of the fan. After the calculation the real speed of the fan was compared to the nominal speed curve from the datasheet provided by the manufacturer (Appendix 1).

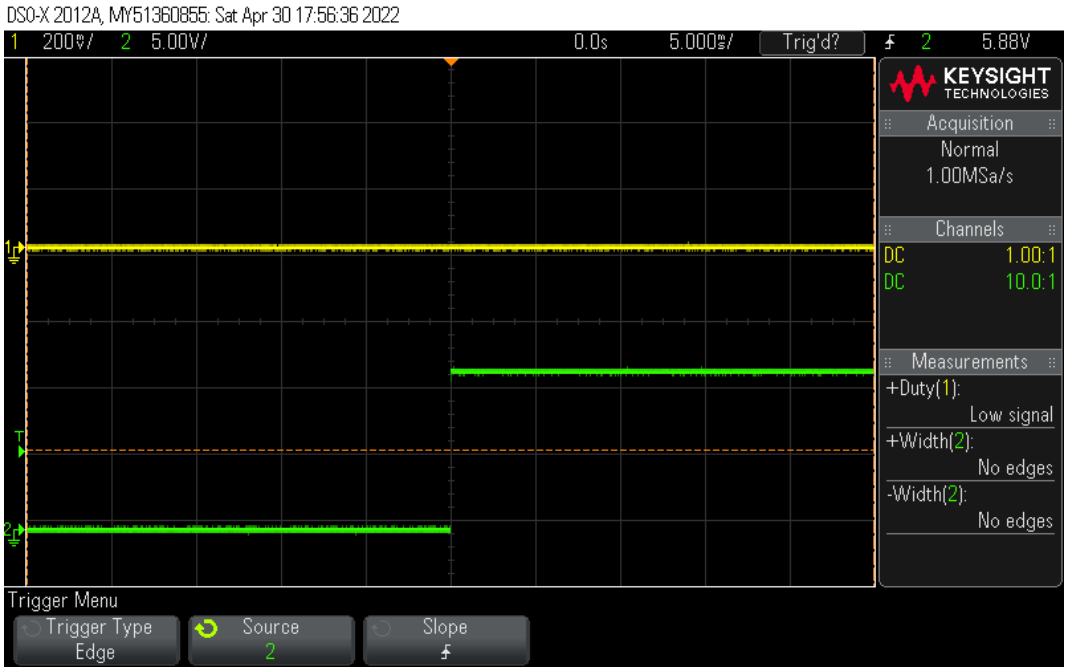


Figure 21. Duty cycle set 0%

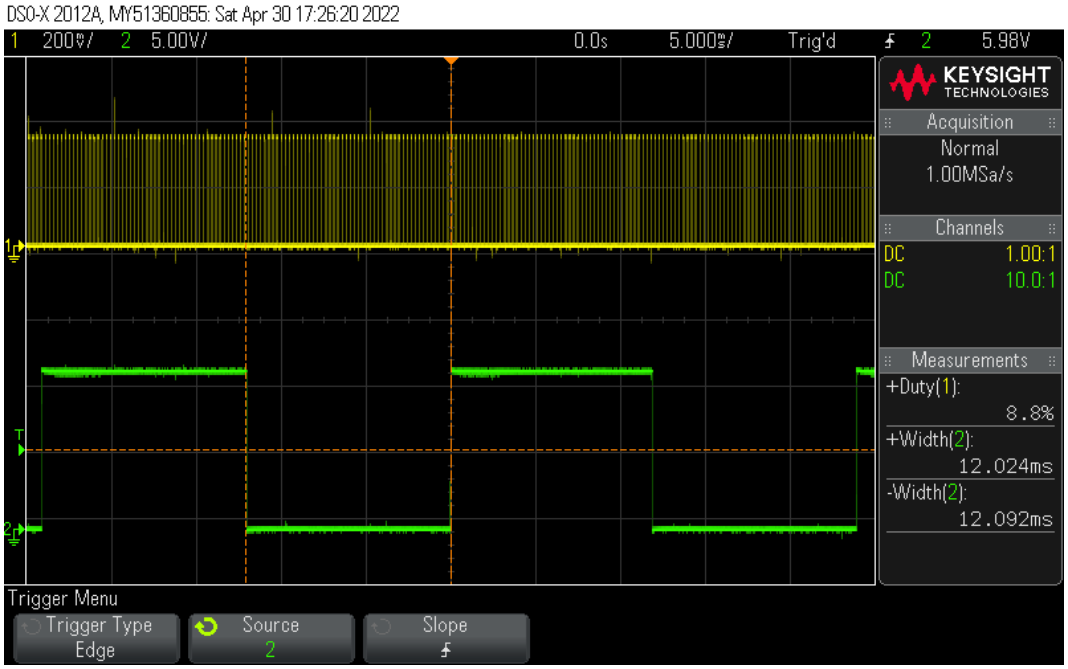


Figure 22. Duty cycle set 10%

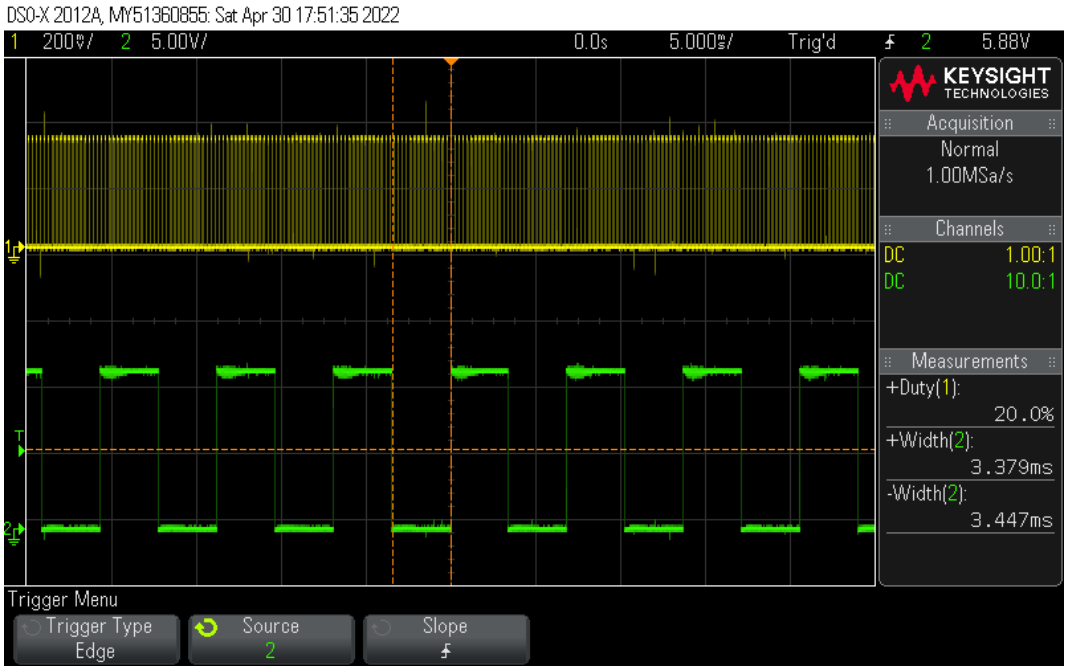


Figure 23. Duty cycle set 20%

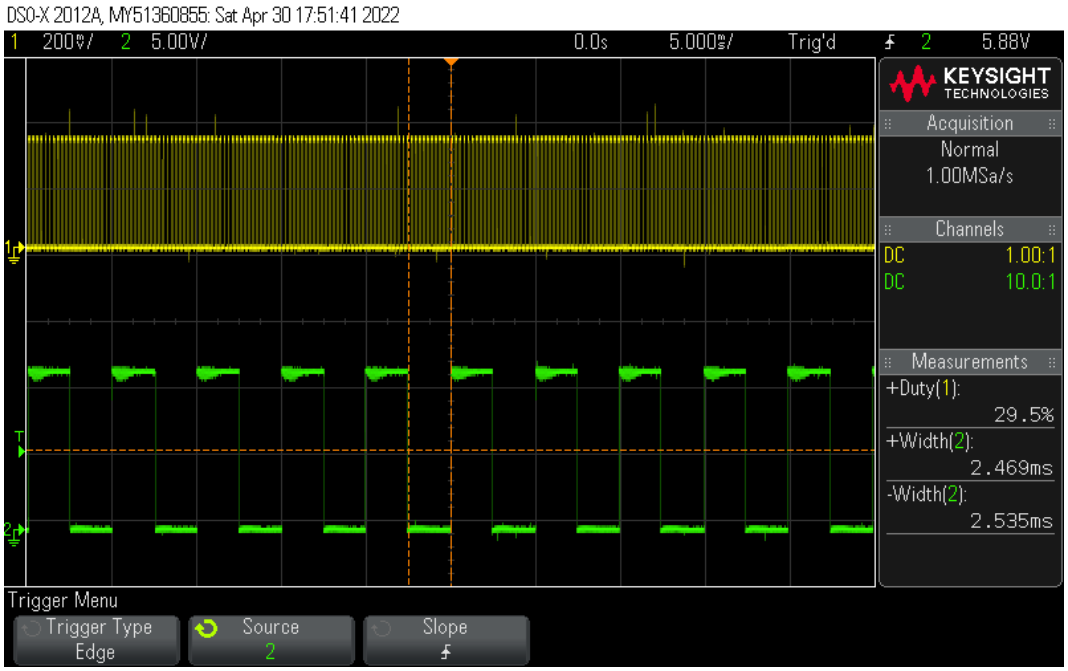


Figure 24. Duty cycle set 30%

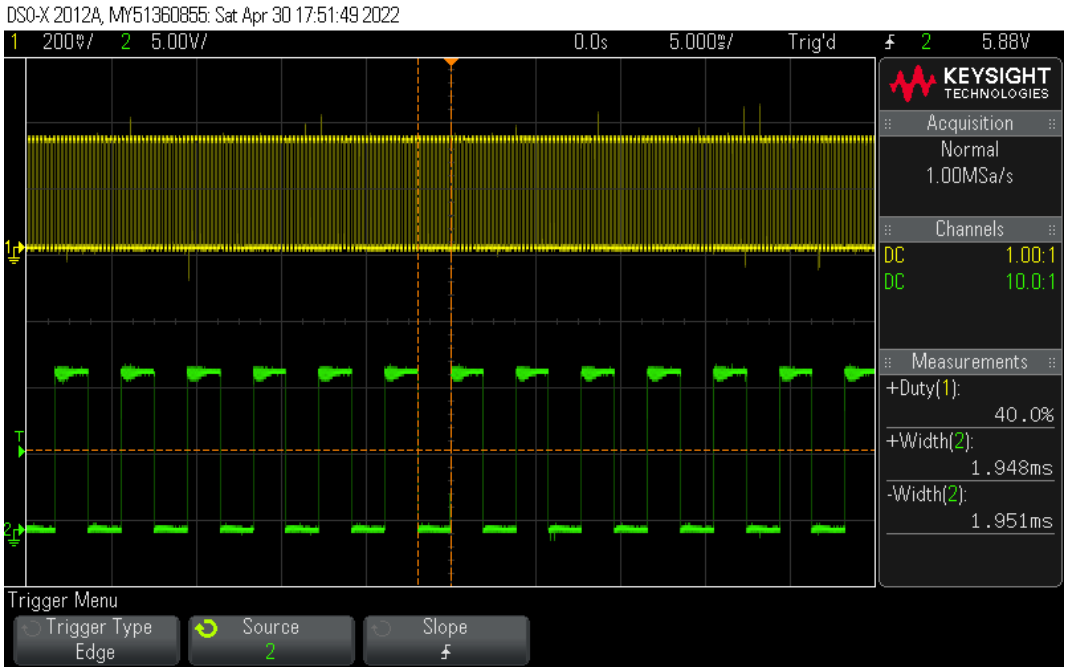


Figure 25. Duty cycle set 40%

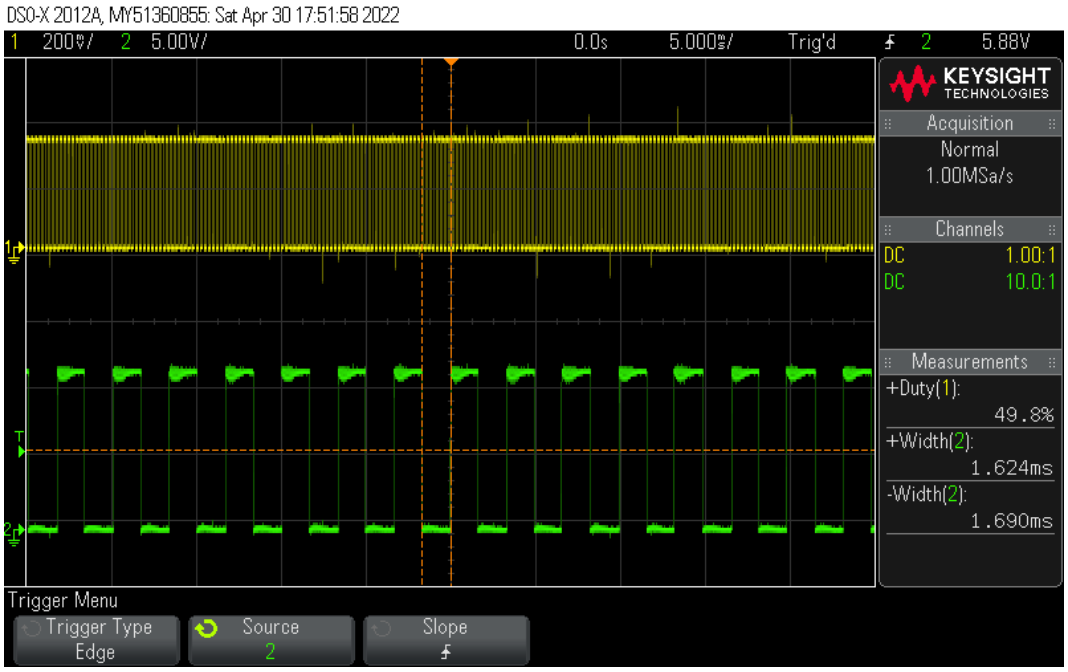


Figure 26. Duty cycle set 50%

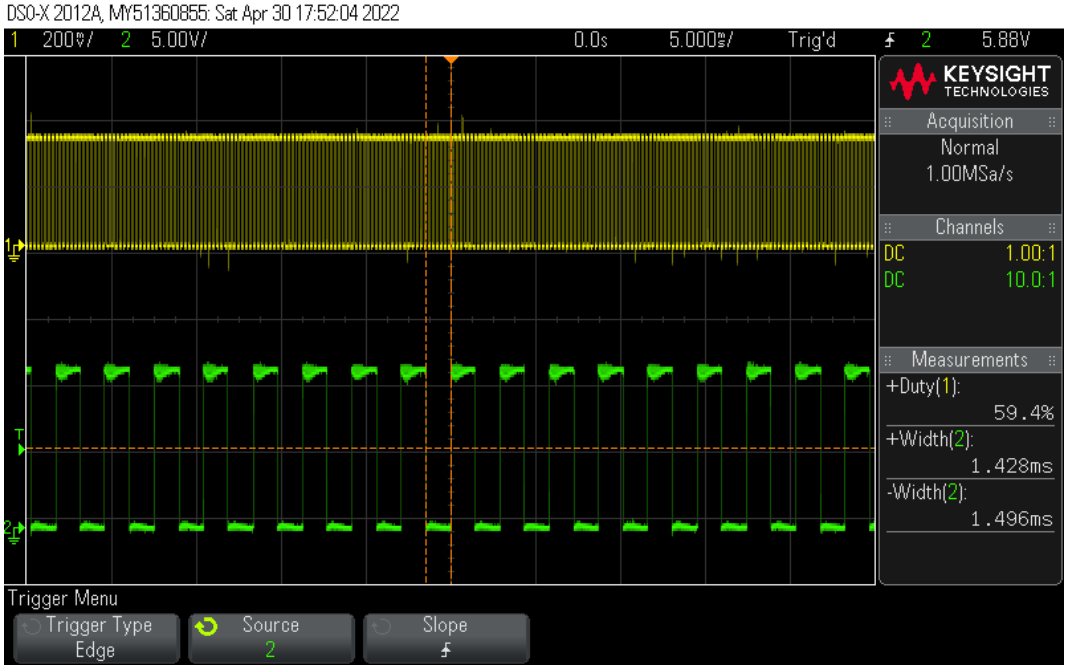


Figure 27. Duty cycle set 60%

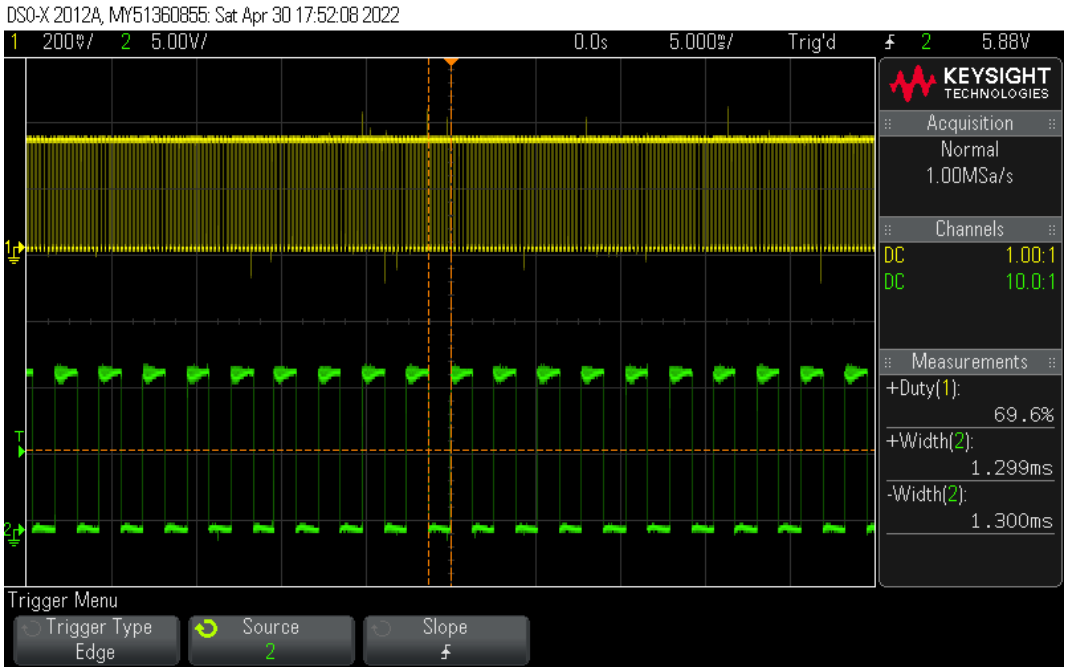


Figure 28. Duty cycle set 70%

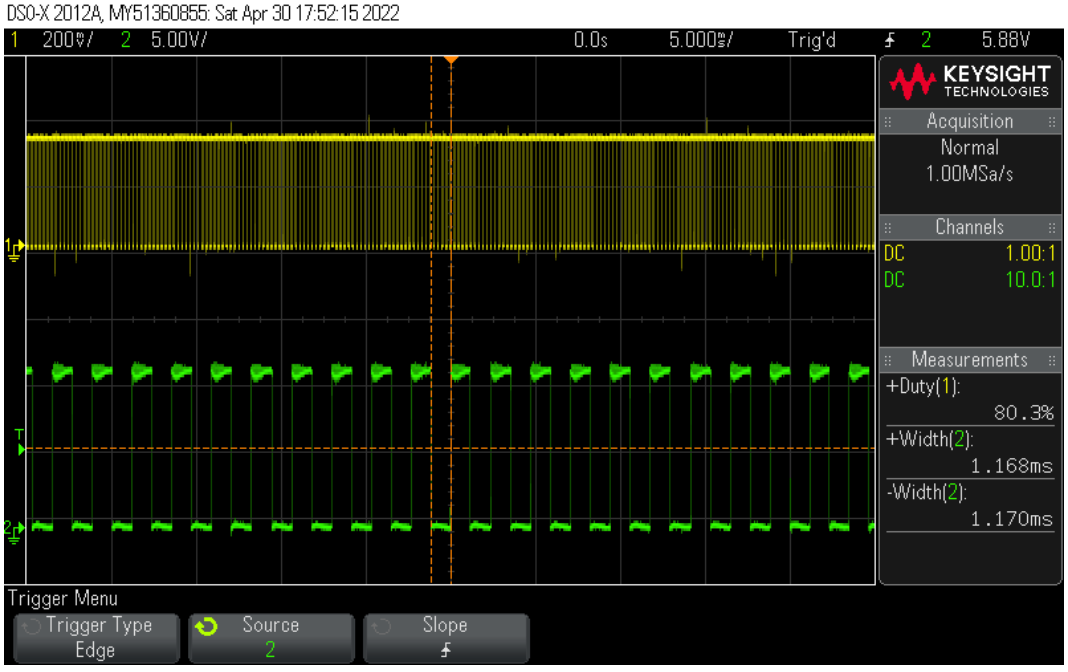


Figure 29. Duty cycle set 80%

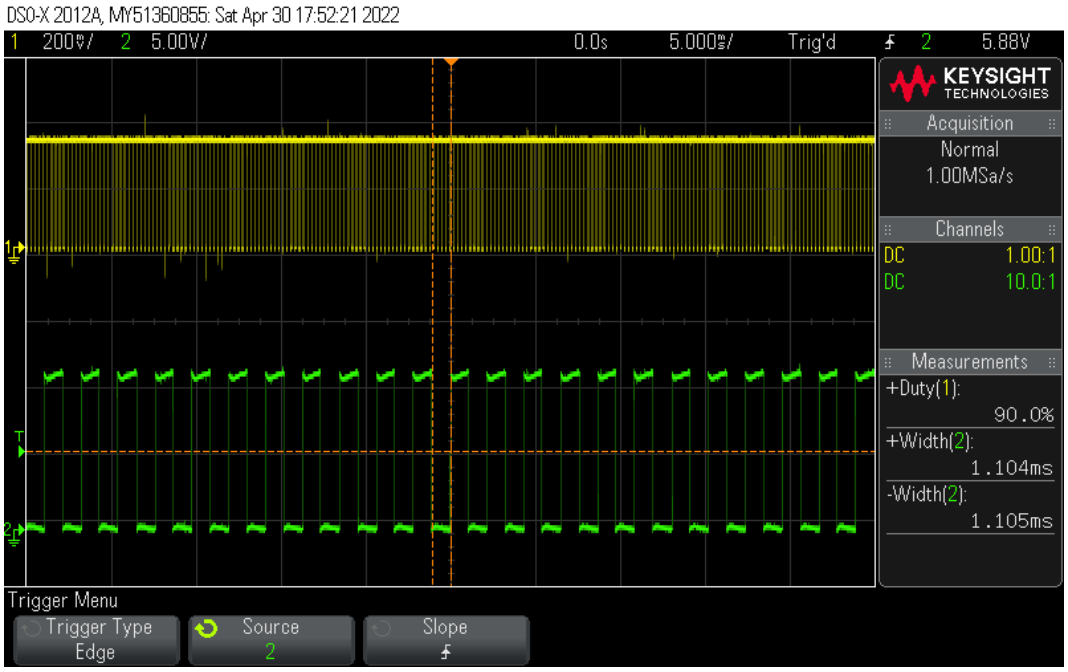


Figure 30. Duty cycle set 90%

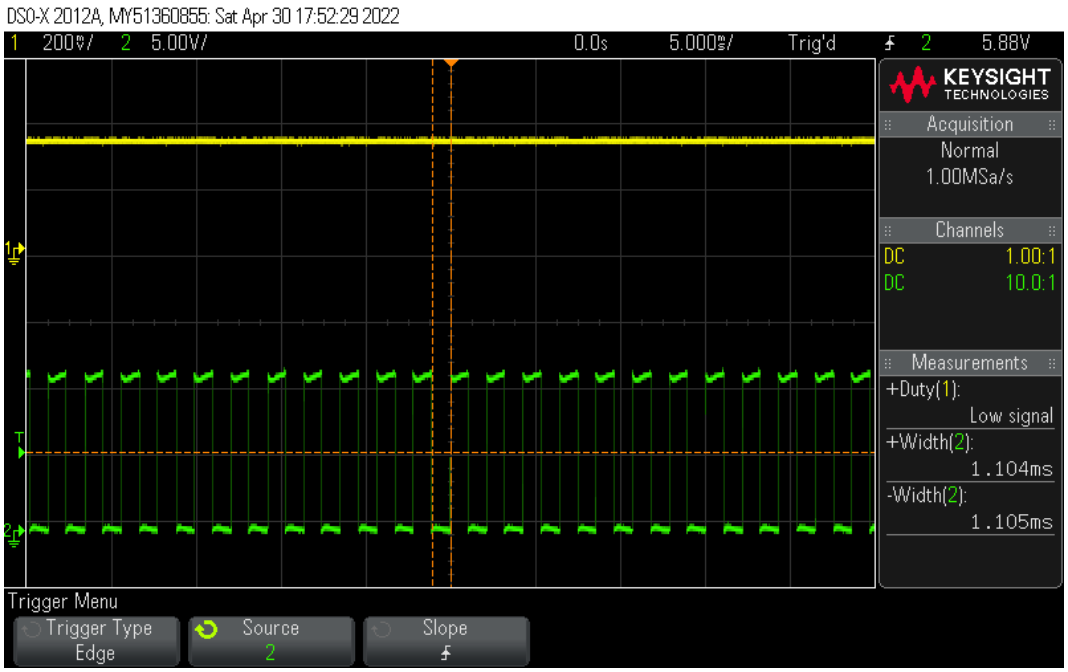


Figure 31. Duty cycle 100% (signal is always on)

Figure 25 shows the case when the trigger is on channel 1, and the oscilloscope cannot measure the period of channel 2. But this clearly shows the signal of the PWM. While performing the test, the speed was sent to the PC terminal at each instance, the received information is shown in Figure 26.

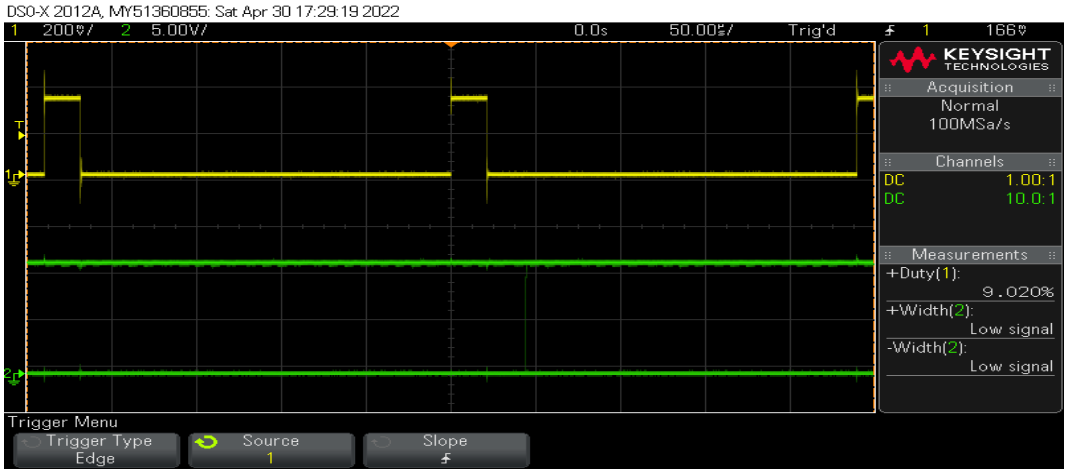


Figure 32. Trigger on PWM

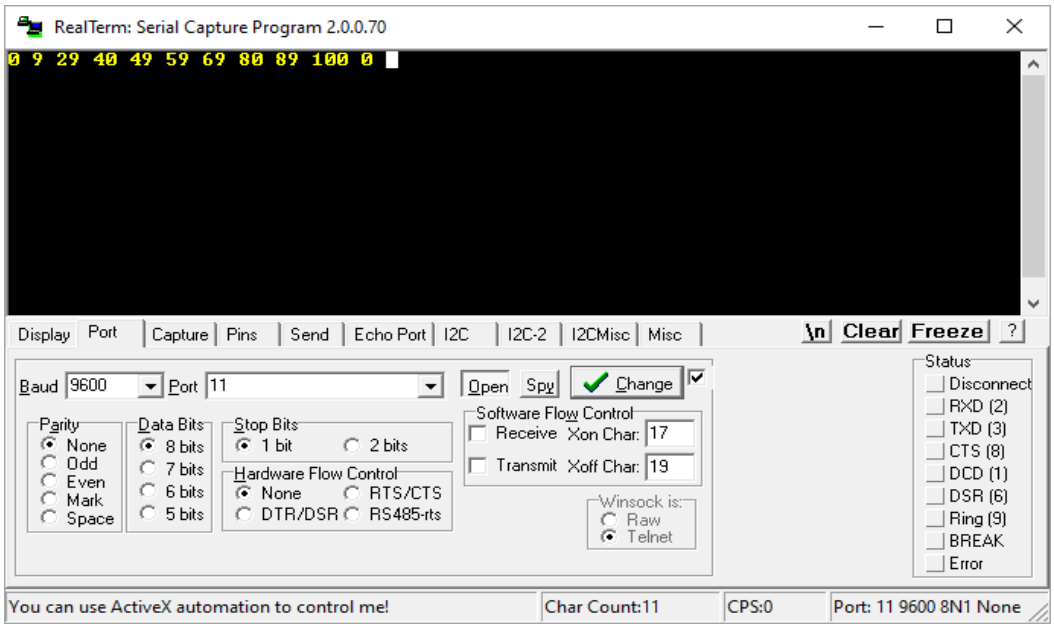


Figure 33. RealTerm window

4 CONCLUSIONS

As a result of the testing the following table was created:

Switches	Duty Cycle (%)	UART	ON time(ms)	Speed (RPM)	Nominal datasheet speed (RPM)
0000	0	0	0	0	0
0001	10	10	12.1	1240	Not specified
0010	20	20	3.4	4412	2450-4550
0011	30	30	2.4	6250	5200
0100	40	40	1.9	7895	6600
0101	50	50	1.6	9375	8000
0110	60	60	1.4	10714	9000
0111	70	70	1.26	11905	10000
1000	80	80	1.16	12931	11000
1001	90	90	1.08	13889	11500
1010	100	100	1.08	13889	10800-12300

Table 2. Testing results

As depicted by the table, the calculated real speed was higher than the nominal values specified by the manufacturer. However, the magnitude of the difference between the calculated value and the nominal are very close at every instance of measurement, therefore it could be the matter of different factors of the environment of the test, such as the source voltage, PWM signal frequency or pullup resistor value.

One obstacle remained after the end of the project, the information that board read from the tachometer was not clear, so it could not be used as feedback, so the system remained open loop.

Nevertheless, the system is proven to be able to control the speed of the fan, and possibly any other device with PWM control accurately and correctly.

REFERENCES

/1/ What is a PID controller? And What is the output of a PID controller? SSLA, 2022. Accessed 15.02.2022. https://www.ssla.co.uk/pid-controller/

/2/ Raginsky M. and Liberzon D., Control systems Lecture 6, 2018. Accessed 16.02.2022. https://courses.engr.illinois.edu/ece486/fa2019/hand-book/lec06.html

/3/ Åström, K. J. & Hägglund, T. 1995 PID Controllers: Theory, Design, and Tuning. 2nd ed. Research Triangle Park, N.C. ISA

/4/ Pedroni, A. V. 2004 Circuit Design with VHDL. 1st ed. Cambridge, Mass.: MIT Press

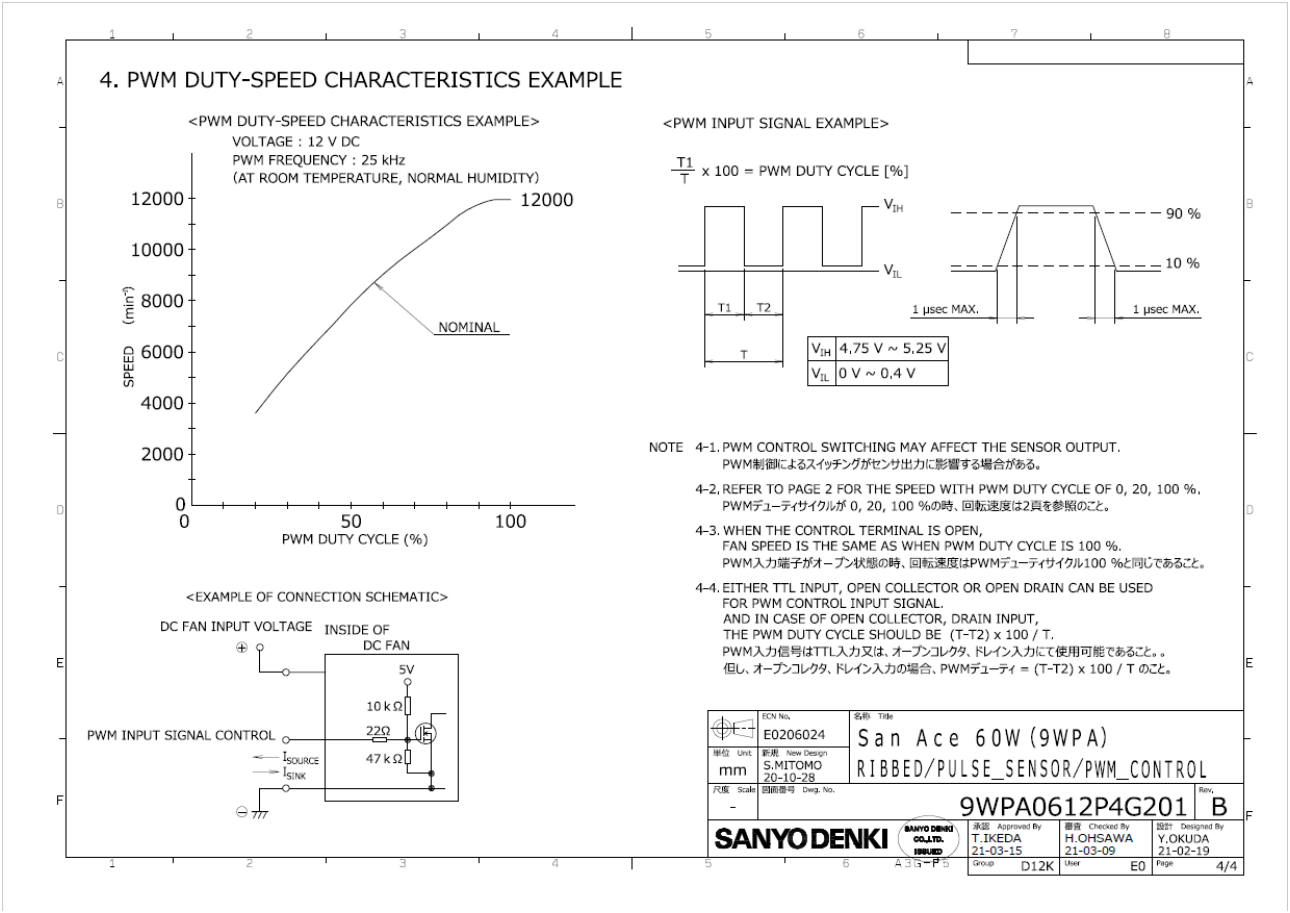
/5/ Heath, J. 2017. Pulse Width Modulation. Accessed 10.03.2022 https://www.analogictips.com/pulse-width-modulation-pwm/

/6/ Trochimiuk, M. 2021. FPGA programming—how it works and where it can be used. Accessed 16.03.2022 https://codilime.com/blog/fpga-programming-how-it-works-and-where-it-can-be-used/

/7/ Cadence PCB solutions. Hardware Description Languages: VHDL vs Verilog, and Their Functional Uses. Accessed 16.03.2022 https://resources.pcb.cadence.com/blog/2020-hardware-description-languages-vhdl-vs-verilog-and-their-functional-uses

APPENDIX 1

Extract from Sanyo Denki 9WPA0612P4G201 datasheet



4. PWM DUTY-SPEED CHARACTERISTICS EXAMPLE

<PWM DUTY-SPEED CHARACTERISTICS EXAMPLE>
VOLTAGE : 12 V DC
PWM FREQUENCY : 25 kHz
(AT ROOM TEMPERATURE, NORMAL HUMIDITY)

<PWM INPUT SIGNAL EXAMPLE>

$\frac{T1}{T}$ x 100 = PWM DUTY CYCLE [%]

V_{IH}	4.75 V ~ 5.25 V
V_{IL}	0 V ~ 0.4 V

NOTE 4-1. PWM CONTROL SWITCHING MAY AFFECT THE SENSOR OUTPUT.
PWM制御によるスイッチングがセンサ出力に影響する場合がある。

4-2. REFER TO PAGE 2 FOR THE SPEED WITH PWM DUTY CYCLE OF 0, 20, 100 %.
PWMデューティサイクルが 0, 20, 100 %の時、回転速度は2頁を参照のこと。

4-3. WHEN THE CONTROL TERMINAL IS OPEN,
FAN SPEED IS THE SAME AS WHEN PWM DUTY CYCLE IS 100 %.
PWM入力端子がオープン状態の時、回転速度はPWMデューティサイクル100 %と同じであること。

4-4. EITHER TTL INPUT, OPEN COLLECTOR OR OPEN DRAIN CAN BE USED
FOR PWM CONTROL INPUT SIGNAL.
AND IN CASE OF OPEN COLLECTOR, DRAIN INPUT,
THE PWM DUTY CYCLE SHOULD BE (T-T2) x 100 / T.
PWM入力信号はTTL入力又は、オープンコレクタ、ドレイン入力にて使用可能であること。。
但し、オープンコレクタ、ドレイン入力の場合、PWMデューティ = (T-T2) x 100 / T のこと。

<EXAMPLE OF CONNECTION SCHEMATIC>

ECN No. E0206024	名称 Title: San Ace 60W (9WPA) RIBBED/PULSE_SENSOR/PWM_CONTROL	
単位 Unit: mm	新規 New Design: S.MITOMO 20-10-28	
尺度 Scale: -	図面番号 Dwg. No.: 9WPA0612P4G201	Rev. B
SANYO DENKI	承認 Approved By: T.IKEDA 21-03-15	審査 Checked By: H.OHSAWA 21-03-09
Group D12K	User E0	設計 Designed By: Y.OKUDA 21-02-19
		Page 4/4