

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter consists of design work for developing an efficient control system algorithm. It starts with explaining the system architecture and hardware of Arrow-bot before focus on software algorithm. In software algorithm design, the bang-bang controller was developed by referring to (Mu`izz, 2011) as a benchmarking performance with the proposed PID algorithm. In the designing of the PID controller algorithm, there is a few approach has been applied that concentrate on sensor data configuration (analog and digital method) and implementing a real-time operating system in order to move the robot accurately and smoothly.

3.2 System Architecture

Block diagram of the system architecture which is shown in Figure 3.1. ATmega328P is the main controller for this design. There are five main categories included ATmega328P for interfacing design, which are basic circuits for microcontroller, IR Sensors, Motor DC, and digital input interfacing and output from LED and push button.

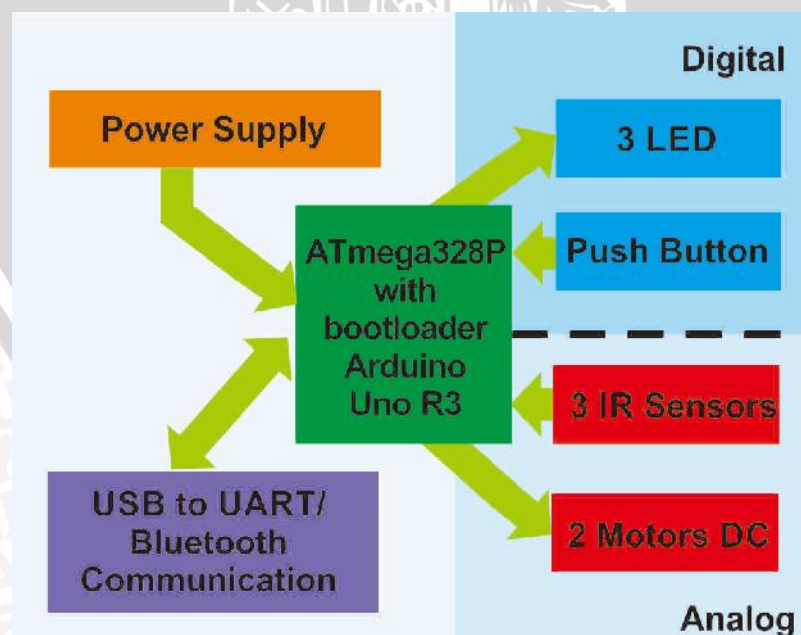


Figure 3.1 Project architecture block diagram.

In basic circuit of microcontroller there are three important circuits that are UART circuit, power supply circuit, sensor circuit, LED circuit, and motor driver circuit. UART circuit is used in propose of boot-loading program from computer to microcontroller and show the data. Power supply circuit is used to distribute the voltage to Arrow-bot. Three reflective optical sensors (IR) send the data to microcontroller as analog input. Microcontroler send PWM signal to motor driver which will be processed to drive the motors. Digital input is push button and digital output is three LEDs as indicator

3.2.2 Block Diagram System

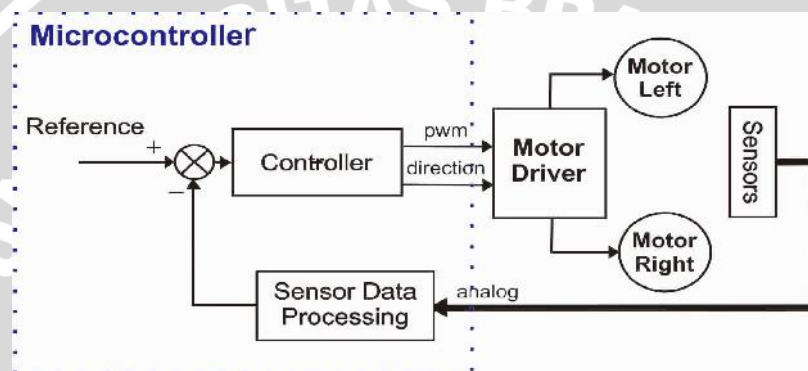


Figure 3.2 Block diagram system.

Set point in this system is divided into two, namely the concept of digital and analog from sensors algorithm. Which in the digital concept the value of the sensors which has been calibrated declared to be two orders (black means 1 and white means 0). ATmega328P with Bootloader Arduino which provides an output signal pulse width modulation (PWM) to the motor driver. In this system used three controllers are bang-bang, PID, PID with real-time operating system. Then the left and right motors as actuators which work alternately as instructed. And as feedback system is used three IR analog sensors to detect the state of the line.

3.2.3 Specification

Because the purpose is to make robot that simple, used for education, small. The specifications are adjusted in accordance with its objectives. Specifications this robot are as follows:

- Range reflective optical sensor readings obtained from the microcontroller of 30 to 120 with 10-bit ADC.
- The dimension of robot is 10cm x 8cm.

- The width of the line will be measured at 1.5 cm.
- The distance between sensors 1 and 3 by 1.5 cm
- The distance between sensors 1 and 2, 2 and 3 by 0.5 cm
- The diameter of wheels is 4.3 cm.
- Weight 150 grams

3.3 Hardware Design

For the hardware design, the printed circuit board (PCB), project components, and the schematic designs will be discussed in detail. Figure 3.3 is an Arrow-bot line follower robot has been created and can be run as desired.



Figure 3.3 Arrow-bot line follower robot overview.

3.3.1 PCB Circuit Board Overview

In this project, there are two printed circuit board (PCB) which is located in the top layer and the bottom layer, which will be described below. Both circuits integrate with each other in a robot that will be created. Thus, the complete range for this project can be obtained.

3.3.1.1 Top Layer PCB

The top layer printed circuit board (PCB) is also referred to as the main controller where the microcontroller circuit board serves as the main component and as the brain which is mounted on the circuit board. Overview components located on the top layer or the main circuit PCB shown in Figure 3.4.

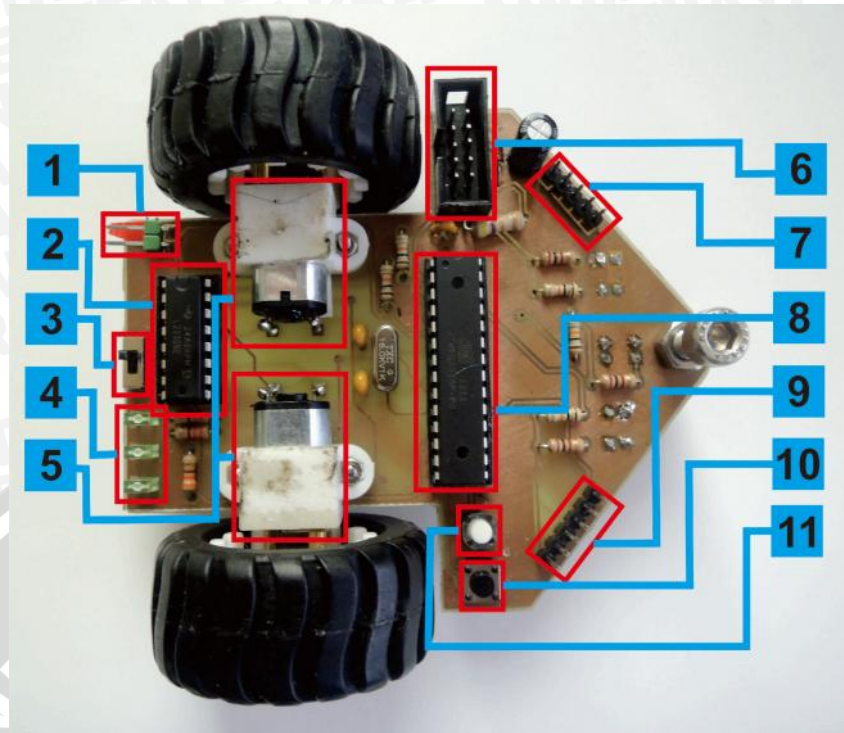


Figure 3.4 Components on the top layer PCB.

Components are shown on the label on the Figure 3.4 above, will be described in the following Table 3.1.

Table 3.1 Components on the top layer PCB.

Label	Component's name and feature
1	5 Voltage Battery Supply
2	L293DNE Motor Driver
3	ON/OFF Toggle Button
4	LED Indicator
5	2 x Motors DC
6	UART Communication
7	J4 Expansion Header (I2C connection)
8	ATmega328P with Bootloader Arduino Uno R3
9	J5 Expansion Header (serial peripheral interface)
10	Start Button
11	Reset Button

3.3.1.2 Bottom Layer PCB

The Bottom layer PCB of only consists of three reflective optical sensors. That the sensors are specifically designed to detect line. Size and distance sensor installation has been previously calculated. Because this bottom layer PCB is designed without a microcontroller on it, the top layer of the PCB is required in the case. Bottom and top

layer integrated with each other. Overview components located in the bottom layer PCB is shown in Figure 3.5.

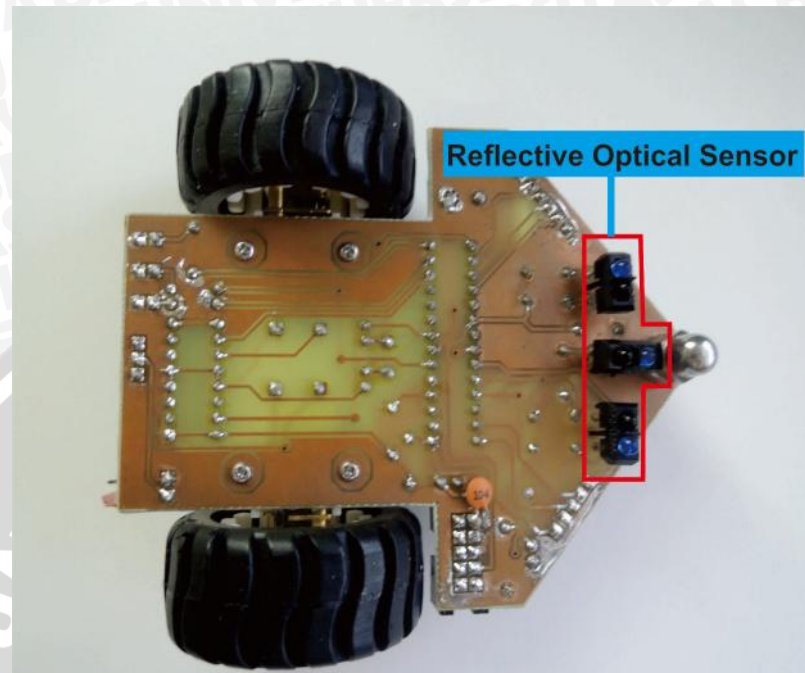


Figure 3.5 Components on the bottom layer PCB.

3.3.2 Project Components

All components detail based on the block diagram design are summarized and listed down in Table 3.2.

Table 3.2 Project component.

Categories	Hardware	Components	Model
Main Control Unit	Microcontroller	Arduino UNO R3	ATmega328P with Bootloader Arduino Uno R3
Input Device	Sensor	3 x Reflective Optical Sensor	Vishay TCRT5000
	Button	Start Button	Push Button 5x5mm
Output Device	Motor Driver	Motors Driver	L293DNE H- Bridge
	Motor	2 x Motor DC	50:1 Micro Metal Gearmotor
	Wheel	2x Wheel	42 mm Diameter
Power Supply	Battery	5 Volt Battery	
Expansion	Expansion Header	6x2 pin 1.27mm (0.05")	
Communication	USB to UART/Bluetooth	USB to UART/Bluetooth	UC00A USB to UART Converter

3.3.2.1 Microcontroller Arduino Uno R3

Arduino Uno is a microcontroller board based on the ATmega328. Arduino Uno has a 14 pin digital inputs and digital outputs (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a button reset. Arduino Uno contains everything needed to support the microcontroller, easy to connect it to a computer with a USB cable or supply it at 5 volts with an AC to DC adapter or battery to get started using. ATmega328 pin configuration is shown in Figure 3.6.

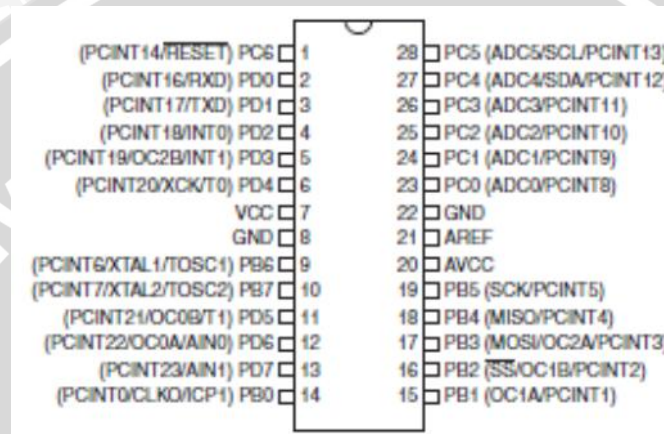


Figure 3.6 ATmega 328 pin configuration.

In this project used an Arduino Uno because consideration of the number of pins used and the size of the robot which is designed as small as possible. Arduino Uno microcontroller suitable as can be removed and installed in the new socket on this robot. So that in this project using chips ATmega 328P from Arduino Uno.

3.3.3.2 Motor Driver L293DNE H- Bridge

DC Motor Driver H-Bridge L293DNE with a DC motor drive circuit H-Bridge is used to control the two PWM DC motor unit and with TTL logic. Circuit DC Motor Driver H-Bridge L293DNE by using L293DNE specially designed as a DC motor driver H-Bridge with 2 units of DC motor control circuit in one IC independent.

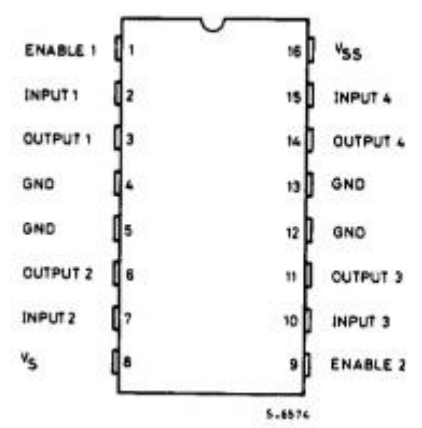


Figure 3.7 IC L293DNE pin configuration.

DC motor driver H-Bridge with L293DNE can be assembled on a small PCB. Motor DC driver H-Bridge with L293DNE can be used to control two DC motors at once. DC Motor Driver H-Bridge with L293DNE also can be used to control a DC motor continuously or with PWM technique.

3.3.3.3 Reflective Optical Sensor

To detect the line in the racing track, most people use two or more reflective optical sensors (Cook, D, 2014) and the outputs of these sensors are fed to comparators with predefined threshold trigger values to get the corresponding digital information. Although this approach works fine for detecting the line, the analog information about the reflection rate of the floor, which is related to whether or not the sensor is right on the line, is lost. As can be seen in reference (Vishay Semiconductor, 2014) This sensor has the following specifications.

Table 3.3 Reflective optical sensor specification (Vishay Semiconductor, 2014).

Description	Value
Voltage	5 Volt
Dimension	10.2x5.8x7mm
Peak operating distance	2.5 mm
Operating range	0.2 mm to 15 mm
Detection	Tape, paper, IBM card, etc



Figure 3.8: Vishay reflective optical sensor (Vishay Semiconductor, 2014).

In the Arrow-bot position sensor placement as shown in Figure 3.9. That is mounted three sensors and distance sensors touching the ground at 0.4 cm. This is ideal because, the specification of the distance sensor operates between 0.2 mm to 15 mm. Although the sensor results in the best sensor at a distance of 2.5 mm.

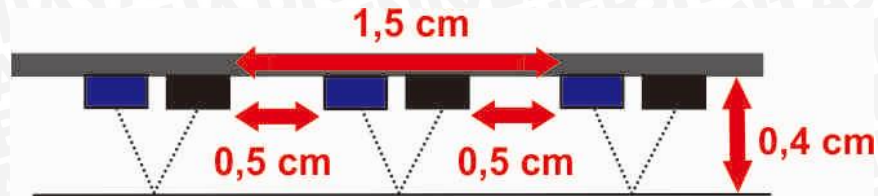


Figure 3.9 Position sensor location on designing Arrow-bot.

3.3.3.4 Motors and Wheels

The movement system is an important part of a robot and its objective is how to move the robot from one point to another point. There are many kinds of motors and wheels. Choice is dependent on Arrow-bot robot function, power, size, grip, and precision.

The motors used in this robot are two motors (50: 1 Micro Metal Gearmotor). The advantages of using this motor are its small size so that in accordance with the purpose of making this robot. On this robot is given to the motor current is equal to 100 mA. Where the voltage is given by 3.01 volts on each motor. DC motors to be controlled have the following specifications.

Table 3.4 Specification motor (Dfrobot, 2015)

Description	Value
Voltage	6 Volt
Rotation Speed	260 RPM
Minimum Current	40 mA
Maximum Current	360 mA
Torque	0.72 Kgcm =0.07 Nm
Weight	10 gram



Figure 3.10 Micro metal gearmotor (Dfrobot, 2015).

A wheel is a circular device that is capable of rotating on a turd through its centre, facilitating movement or transportation while supporting a load (mass), or performing labour in machines. Common examples are found in transport applications. A wheel, together with an axle overcomes friction by facilitating motion by rolling. This robot is used two wheels. The diameter size of the wheel that used in this project is 42 mm and has a weight 16 gram.



Figure 3.11 Wheel.

3.3.2 Schematic Design

In this schematic phase, all related components that have been studied and suitable circuit will be designed based on each components specifications. Schematic will be discussed as below.

3.3.2.1 Main Controler Unit Circuit

ATmega328P with bootloader Arduino Uno R3 as the main processing center in the control process. I / O configuration of the ATmega328P with bootloader Arduino Uno R3 is shown in Figure 3.12. From this figure shows the robot using 21 pins which consists of 5 analog pins, 13 digital pins, and 1 RST pin. On this robot just taken ATmega328P chips because as a consideration to minimize the size of the robot. In AVCC pin coupling capacitor for stabilizing the microcontroller power.

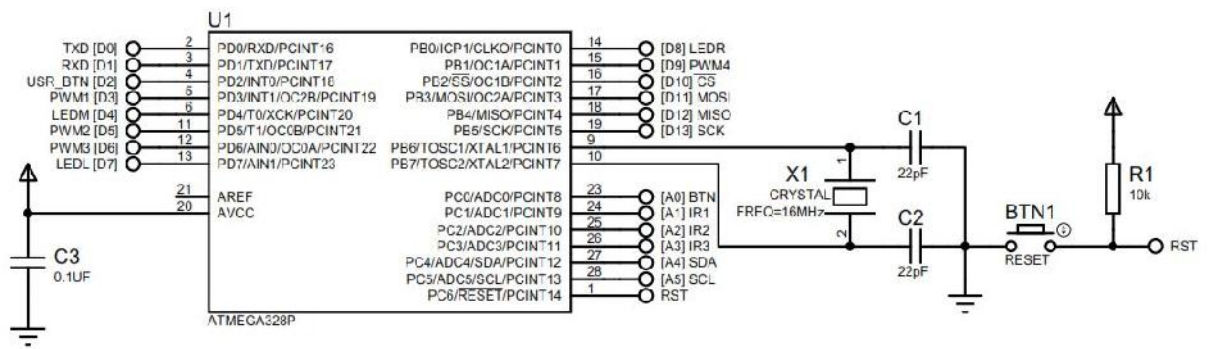


Figure 3.12 Schematic design I/O Arduino Uno.

Table 3.5 Arduino Uno pin function.

No.	Pin	Function
1	A0	BTN (Used as expansion header)
2	A1	Analog input from IR sensor left (1)
3	A2	Analog input from IR sensor middle (2)
4	A3	Analog input from IR sensor right (3)
5	A4	SDA (Used as expansion header)
6	A5	SCL (Used as expansion header)
7	DO	TXD (Used as an UART connection)
8	D1	RXD (Used as an UART connection)
9	D2	USR_BTN
10	D3	Used as an interface driver motor left (PWM 1)
11	D4	Turning on and off LED middle
12	D5	Used as an interface driver motor left (PWM 2)
13	D6	Used as an interface driver motor right (PWM 3)
14	D7	Turning on and off LED left
15	D8	Turning on and off LED Right
16	D9	Used as an interface driver motor right (PWM 4)
17	D10	CS (Used as expansion header)
18	D11	MOSI (Used as expansion header)
19	D12	MISO (Used as expansion header)
20	D13	SCK (Used as expansion header)
21	RST	RST (Used as an UART connection)

3.3.2.2 Motor Driver and Motors Circuit

To operate the circuit motor driver L293DNE H- Bridge is to provide with PWM pulses at the input terminals. The terminals are IN1- IN2 IN3 - IN4. Enable input lines 1 and 2 are used to enable DC motor and given 5 Volt. Circuit motor driver

L293DNE H- Bridge on Arrow-bot can be seen in the Figure 3.13 and the explanation shown in Table 3.6 below.

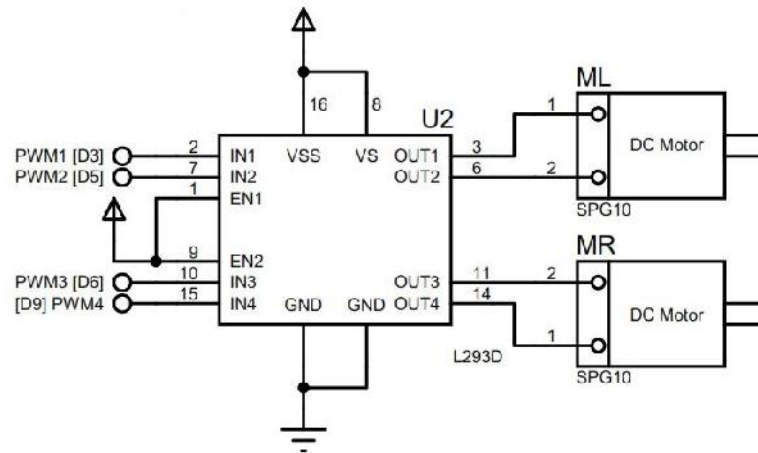


Figure 3.13 Schematic motor driver and motors.

Table 3.6 Motor driver pin function.

No.	Pin	Function
1	IN1	Input from PWM 1 Arduino
2	IN2	Input from PWM 2 Arduino
3	EN1	Always Active (connected to 5 V)
4	EN2	Always Active (connected to 5 V)
5	IN3	Input from PWM 3 Arduino
6	IN4	Input from PWM 4 Arduino
7	OUT1	Output PWM (Left Motor Clockwise)
8	OUT2	Output PWM (Left Motor Anti Clockwise)
9	OUT3	Output PWM (Right Motor Clockwise)
10	OUT4	Output PWM (Right Motor Anti Clockwise)
11	GND	Grounding
12	VSS	Conncted to 5V
13	VS	Connected to 5V

3.3.2.3 LED Configuration Circuit

Three LED are used as indicator of this robot. The principle of making the circuit in this project is before the 3 LEDs connected to pin ATmega328P, given by 330 ohm resistors which serves to reduce the incoming flow of ATmega328P with bootloader Arduino Uno R3. So that the LEDs are safe and functioning properly. Schematic will be described in the Figure 3.14 below.

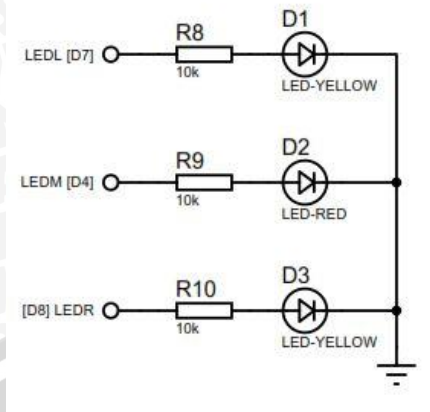


Figure 3.14 Schematic LED configuration.

3.3.2.4 Sensors Configuration Circuit

In the circuit configuration of the sensor on the pin 4 wherein the sensor towards the ATmega328P in parallel with a resistor of 10 kilo ohms heading 5 Volt power supply. Pin 3 and 2 go to ground. At one pole are given resistor of 100 ohm and headed to the 5 volt power supply. Because one pin is used as UART that requires a power supply of 3.3 volts, the output voltage 5 V voltage to be reduced with diodes, capacitors and resistors on the circuit as Figure 3.15 below.

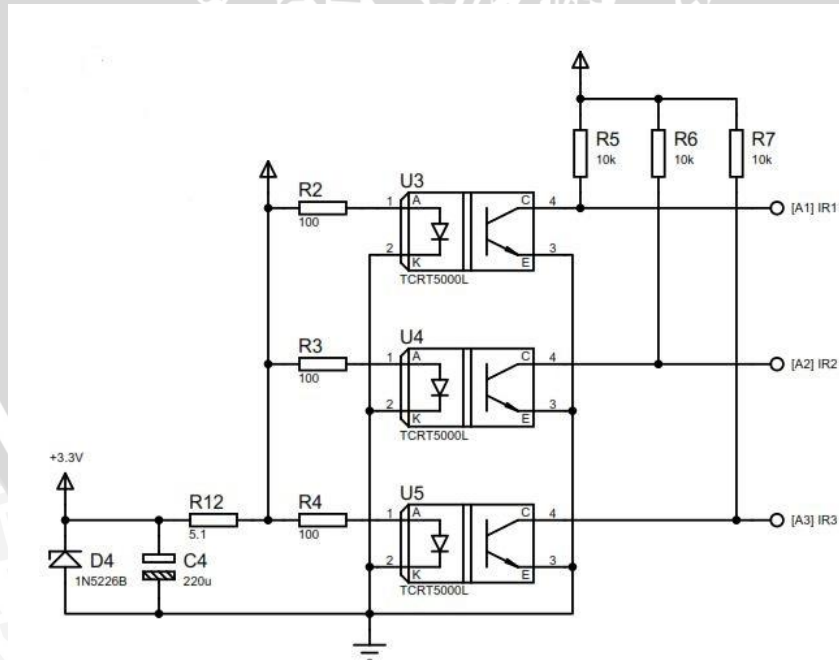


Figure 3.15 Schematic sensors configuration.

3.3.2.5 Power Supply and UART Circuit

Power supply is equal to 5 volts, the reason is to make this robot to its smaller size, thereby reducing component used because of ATmega328P with bootloader Arduino Uno R3 using 5 volt voltage and the motors are also given the same voltage. In schematic Figure 3.16 shows at the Vcc power supply is connected to the switch.

In UART consists of pin RXD, TXD, RST from ATmega328P pins. Ground pin is connected directly to the ground in Battery. As a digital input pin the push button pole one is connected to ground and pole 2 before being connected to Vcc given a 10k ohm resistor and then connected to a digital input (D2) on the Arduino Uno bootloader ATmega328P with R3.

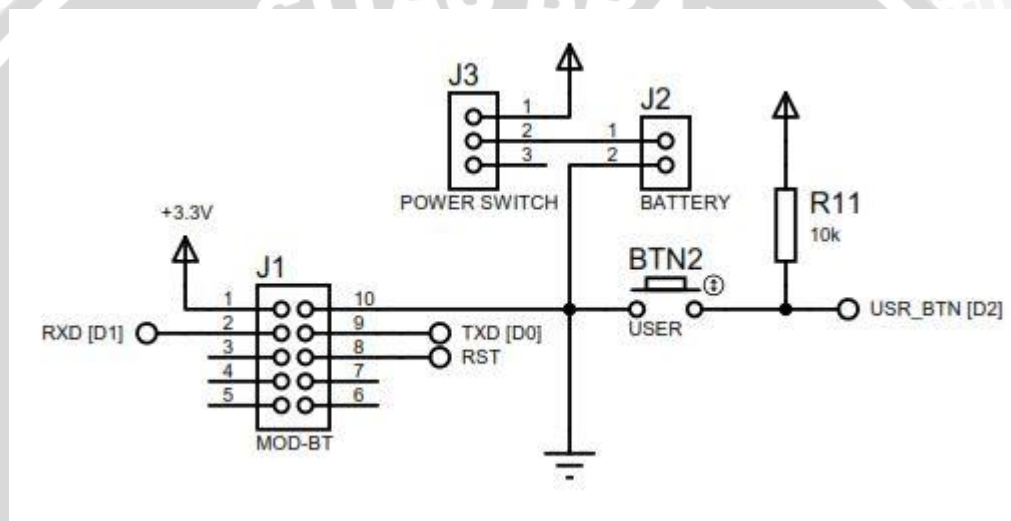


Figure 3.16 Power supply and UART schematic.

3.3.2.6 Expansion Circuit

Expansion header on the robot is divided into two parts, namely J4 and J5. At J4 consists of 5 Volt Vcc pin, ground pin, SCL pin, SDA pin and BTN pin. At J5 consist of 5 Volt Vcc pin, ground pin, CS pin, MOSI pin, MISO pin, SCK pin. J4 has a function as I2C connection and J5 as serial peripheral interface. J4 can be used as an expansion, such as the addition of an accelerometer sensor, gyrometer, compas, and others.

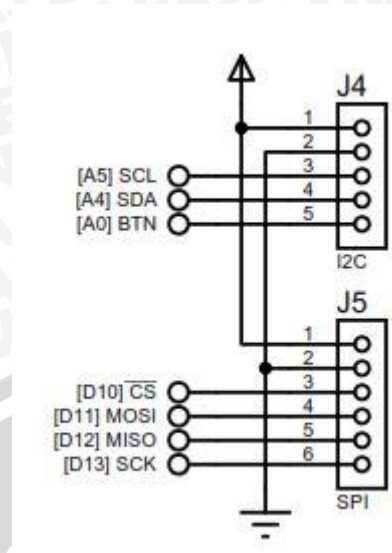


Figure 3.17 Expansion schematic.

3.4 Testing Hardware

The purpose of this testing hardware is to determine whether the tool has been created to function properly and in accordance with the design. In addition, testing is also used as a calculation algorithm in the making of this robot. Testing on this system includes testing each block and the overall test. Testing each block is performed to locate faults and simplify the analysis of the system if the device does not work in accordance with the design.

3.4.1 Calibration Reflective Optical Sensor

To detect the line to be followed, most people use two or more reflective optical sensors, and the outputs of these sensors are fed to comparators with predefined threshold values to get the corresponding digital information. Although this approach works fine for detecting the line, the analog information about the reflection rate of the floor which is related to whether or not the sensor is right on the line is lost (Dudek, and Jenkin, 2000).

The purpose of calibration this sensors was to determine the ability of the sensor IR to reading discoloration. In this experiment using two sample colors: black and white. Because that is used in this robot is reading black and white. The data obtained by comparing the value from Atmega 328P with a ratio of the actual color. Test data will be shown in the following Table 3.7.

Table 3.7 The reflective optical sensor outputs for before normalization.

ADC Output	Colors
Result \leq 30	White
$30 <$ Result $<$ 150	Between White and Black
Result \geq 150	Black

In Table 3.7 above shows that the 100% white color has a range of 0-30 and 100% black color also has a range of 150 to more than that. Where the difference lies in the value of various factors such as a lighter color, different characteristics, or light intensity. It is necessary to do the normalization in the determination the exact value by put limits on the value of black and white. So that the analog output values can then be used to determine the correct position of the line and easy to programming the algorithm. Below is a Table 3.8 showing the data after normalization.

Table 3.8 The reflective optical sensor outputs for after normalization

ADC Output	Colours
30	White
$>30, <$ 150	White Until Black
150	Black

3.4.2 Testing Driver Motor

This testing aims to determine the performance and the response of the motor driver circuit controlling L293DNE H-Bridge by comparing and testing the response signal from the motor driver of the PWM input signal given by the ATmega328P.

The equipments used are L293DNE H-Bridge driver, Oscilloscope, and ATmega328P. Oscilloscope connected to the input and output L293DNE H-Bridge driver. Atmega328P provide instruction signal input on pin PWM motor driver L293DNE H-Bridge with 0-100% duty cycle. Then the data from Oscilloscope is recorded and compared. Large duty cycle ratio between driver input with a duty cycle of the output drivers can be seen in Table 3.9 and Table 3.10.

Table 3.9 Testing results L293DNE H-Bridge driver (left motor).

<i>Duty Cycle Input (%)</i>	<i>Duty Cycle Output Driver (%)</i>	<i>Error (%)</i>
100	100	0
90.1	90.1	0
80.3	80.3	0
70.3	70.3	0
60.2	60.2	0
50	50	0
39.9	39.9	0
30	30	0
20	20	0
9.9	9.9	0
Average Error		0.26

Table 3.10 Testing results L293DNE H-Bridge driver (right motor).

<i>Duty Cycle Input (%)</i>	<i>Duty Cycle Output Driver (%)</i>	<i>Error (%)</i>
100	100	0
90.2	90.2	0
80.2	80.2	0
70.2	70.2	0
60.2	60.2	0
50.3	50.3	0
39.9	40	0.1
30.2	30.2	0
20.1	20.1	0
10	10	0
Average Error		0.01

Based on Table 3.9 and 3.10 can be concluded L293DNE H-Bridge driver can work very well because the average error is generated 0% on the left motor and 0.01% on the right motor. Graph the relationship between the duty cycle of the input to the output duty cycle L293DNE H-Bridge driver from Table 3.9 and 3.10 can be seen in Figure 3.18 and 3.19.

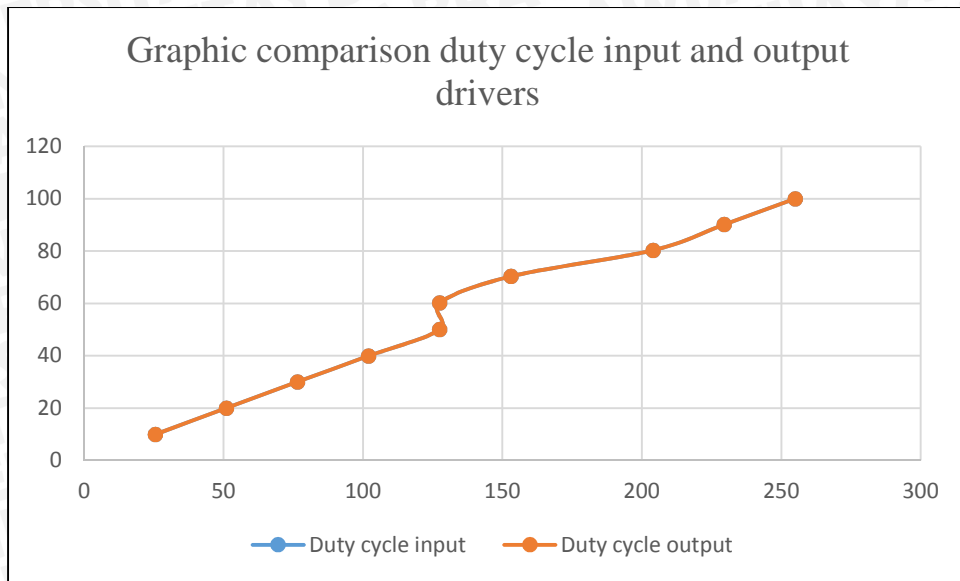


Figure 3.18 Graphic comparison duty cycle input and output driver (left motor).

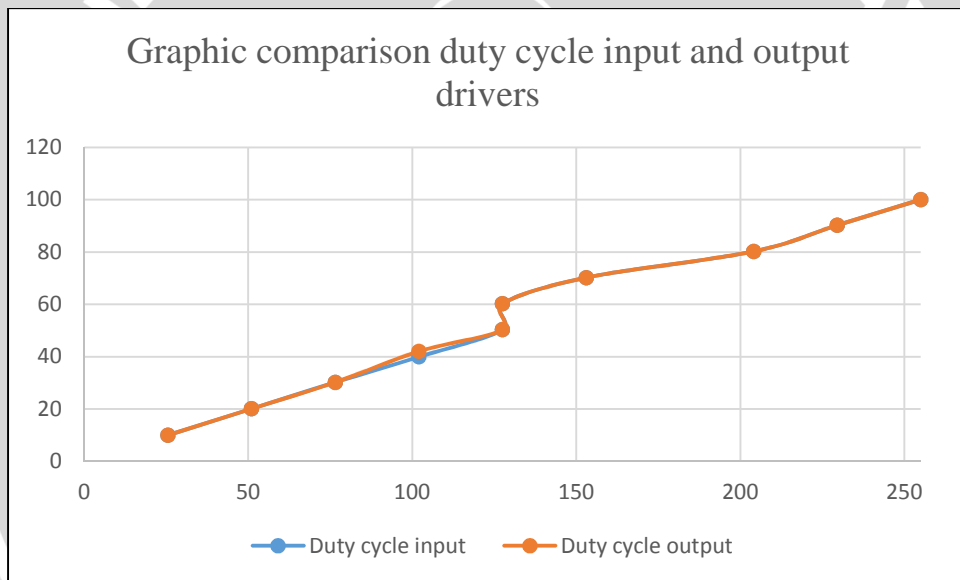
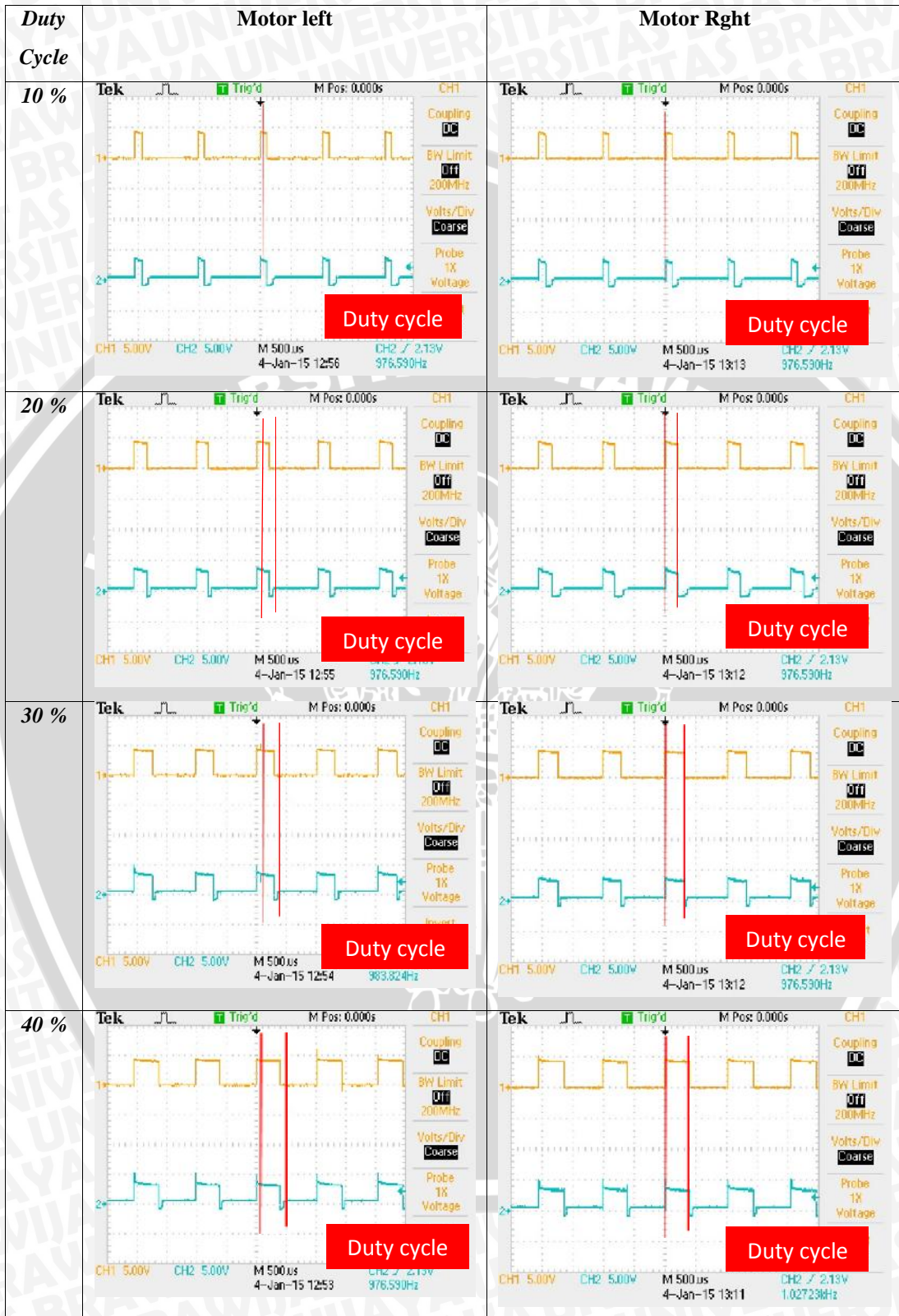
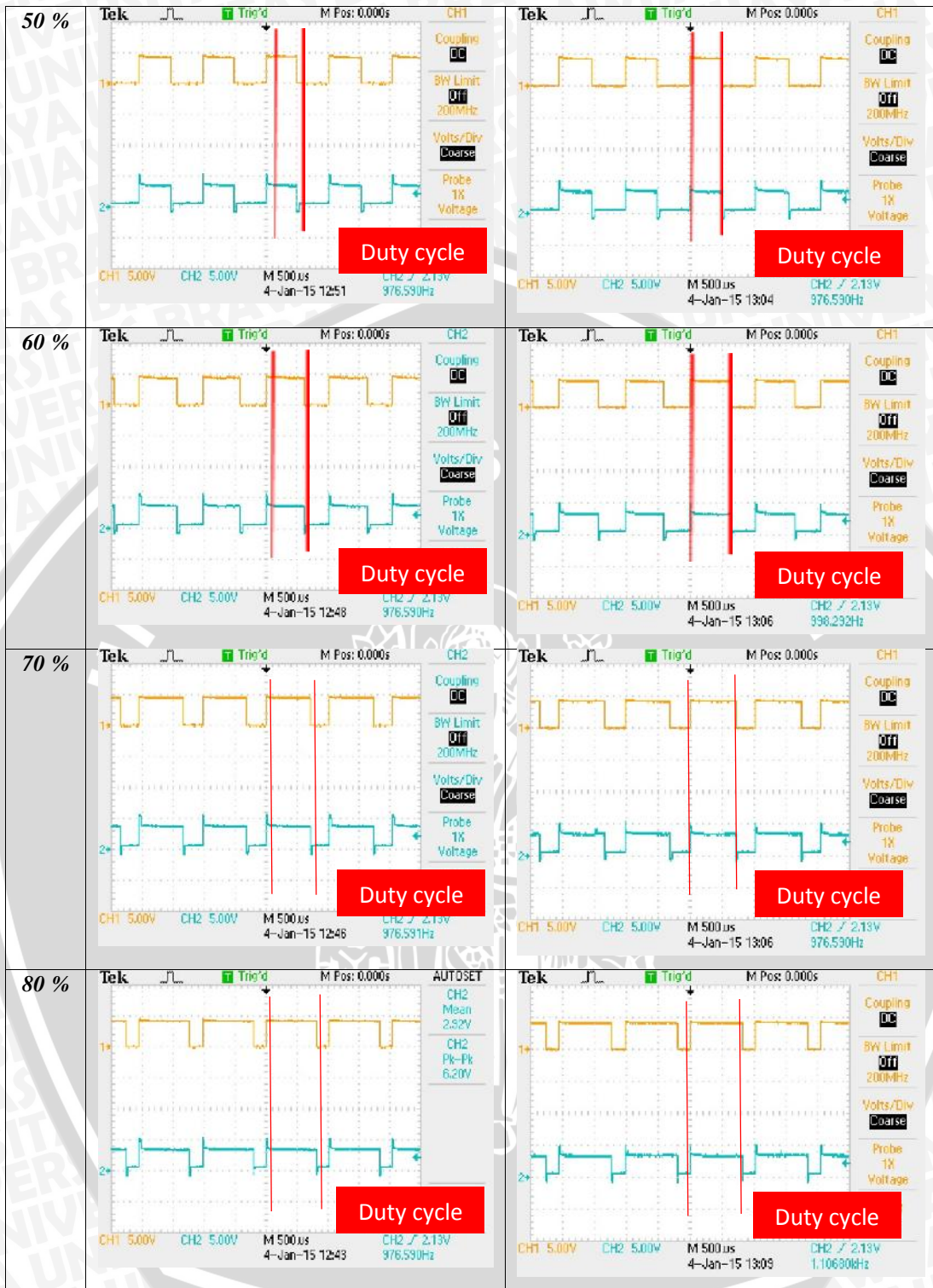


Figure 3.19 Graphic comparison duty cycle input dan output driver (right motor).

The test results PWM signal and the output motor driver L293DNE h-bridge driver using oscilloscope and display program with a duty cycle of the waveform parameters are shown in Table 3.11.

Table 3.11 Signal output and input response L293DNE h-bridge driver 30a and against duty cycle (orange line is input and blue line is output).





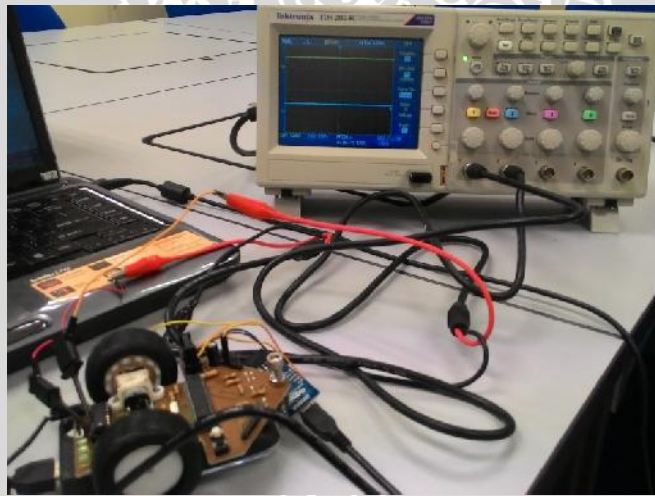
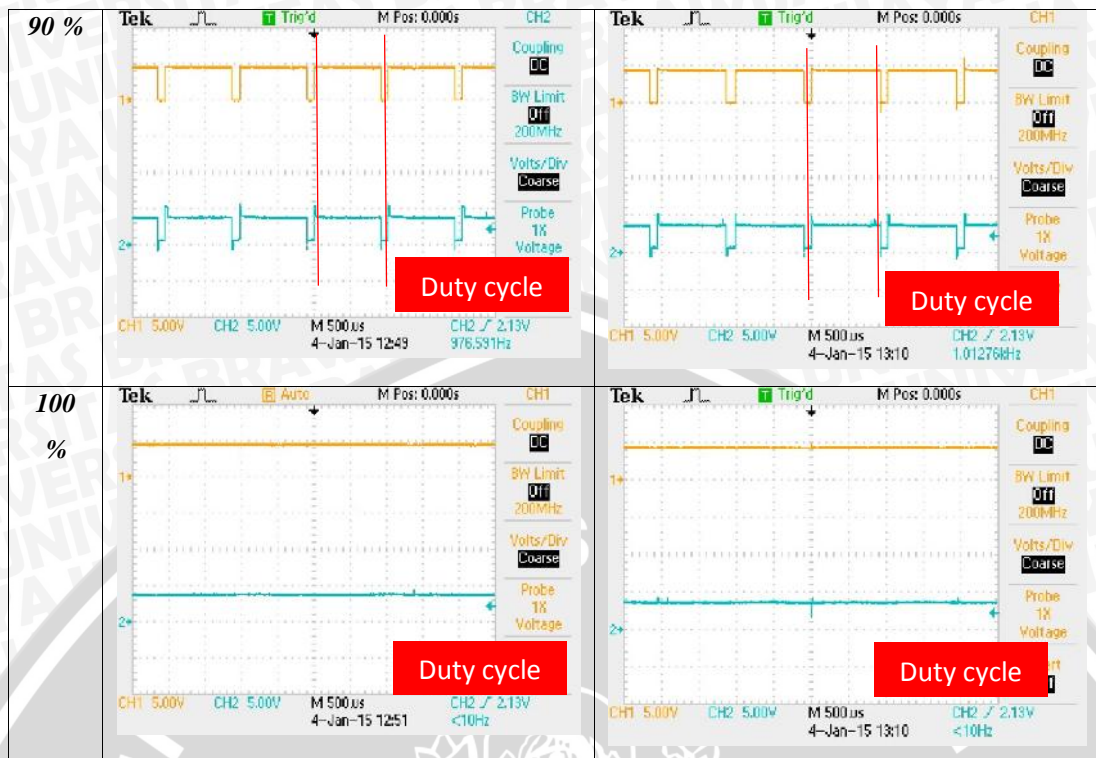


Figure 3.20 Test duty cycle using digital oscilloscope.

3.4.3 Motor DC

DC motor control using ATmega328P with bootloader Arduino Uno R3 as a processing circuit and provide pulse width modulation (PWM) so that the motor can move. Two DC motors are used in this design is not well-characterized, so that needs to test using digital tachometer. The characteristics of 2 DC motors in this design is obtained by giving a unit step input. The results of the PWM motor speed is shown in Figure 3.21 and Figure 3.22.

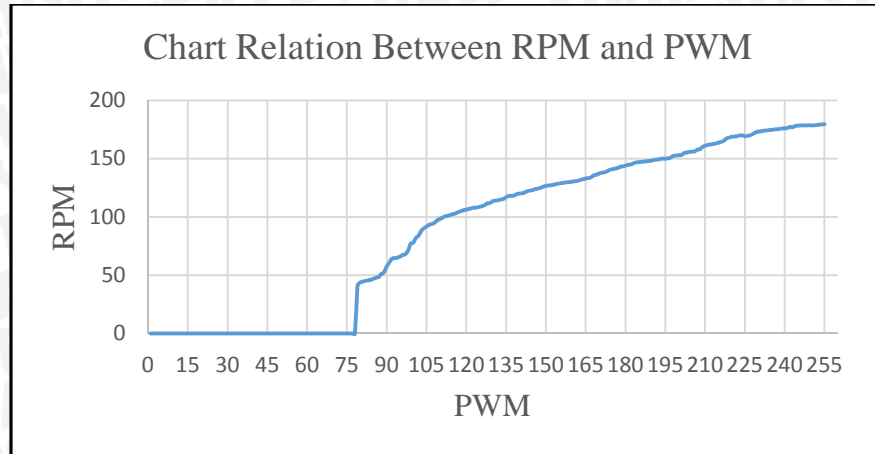


Figure 3.21 Graphic characteristics of DC motor right.

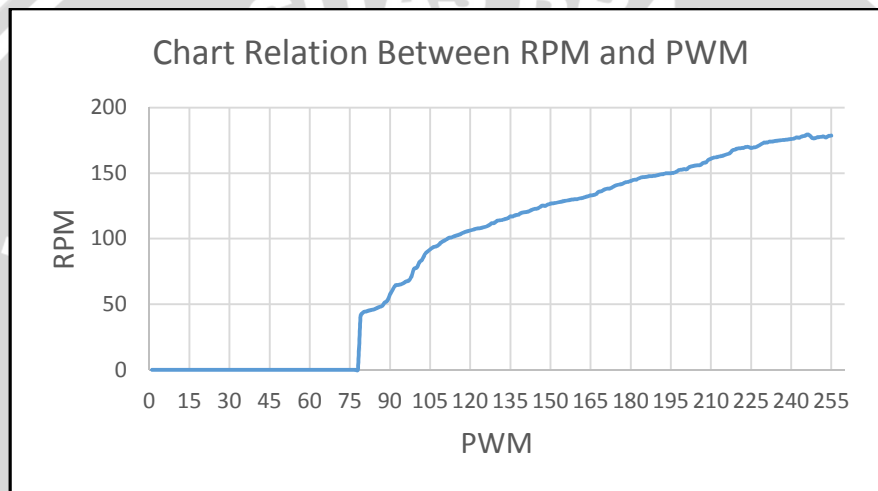


Figure 3.22 Graphic characteristics of DC motor left.

From the data in the image above explains that both the motor started working on PWM 77-99 or 30% -38.8% of the duty cycle. At this DC motor using a power supply is 3.71 Volts and the current is 70 mA.



Figure 3.23 Test motor speed using digital tachometer.

DC motors that will be controlled have rotate speed between 0-180 RPM. The magnitude of the speed regulated by varying the input voltage. DC motor parameters as in the following table.

Table 3.12 Parameters of motor 1 and 2.

Parameters	Motor 1 and 2
Armature Resistance R_a ()	1
Armature Inductance L_a (H)	0.03
Moment of Inertia J ($\text{Kg} \cdot \text{m}^2$)	0,93
Friction Constant B (Nms)	1
Torque Constant K_T (Nm/A)	1.02
EMF Constant K_b (V/s.rad)	1.2

So if included in the equation 2.11, motor transfer function.

$$\frac{\omega(s)}{T_m(s)} = \frac{K_T}{(L_a(s)+R_a)(J(s)+B)+K_b K_T} \quad (\text{Eq. 3.1})$$

Values in the above table included in the calculation of 3.1,

$$\frac{\omega(s)}{T_m(s)} = \frac{(1.02)}{(0.03s+1)(0.93s+1)+1,2 \cdot 1,02} \quad (\text{Eq. 3.2})$$

So that the motor transfer function is :

$$F(s) = \frac{\omega(s)}{T_m(s)} = \frac{1.02}{0.0093 s^2+0.94s+1.224} \quad (\text{Eq 3.3})$$

3.5 Software Design

The overall design of the software is divided into three, namely using a bang – bang controller, using the PID controller, and using the PID controller with RTOS ChibiOS. From the Figure 3.24 describes flowchart system software using bang- bang controller development begins with the start button and then the process begins sensor readings, then from the sensor readings directly determined how the PWM is needed when one sensor is active and not active.

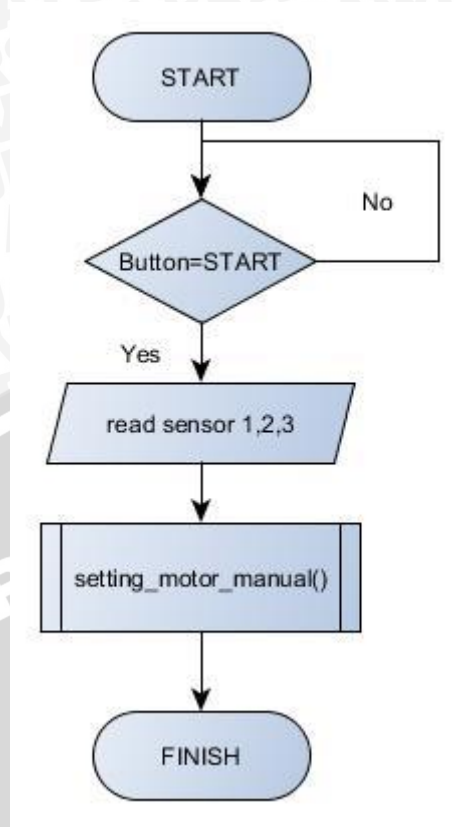


Figure 3.24 Flowchart overall system using bang-bang controller.

From the Figure 3.25 describes flowchart system software using PID controller. Development begins with the start button and then the process sensor readings, if the sensor showed a white line the Arrow Bot line follower will stop, when the active sensor detects black line, sensor readings will be calculated error value. Error value as a reference for the PWM drive the motor.

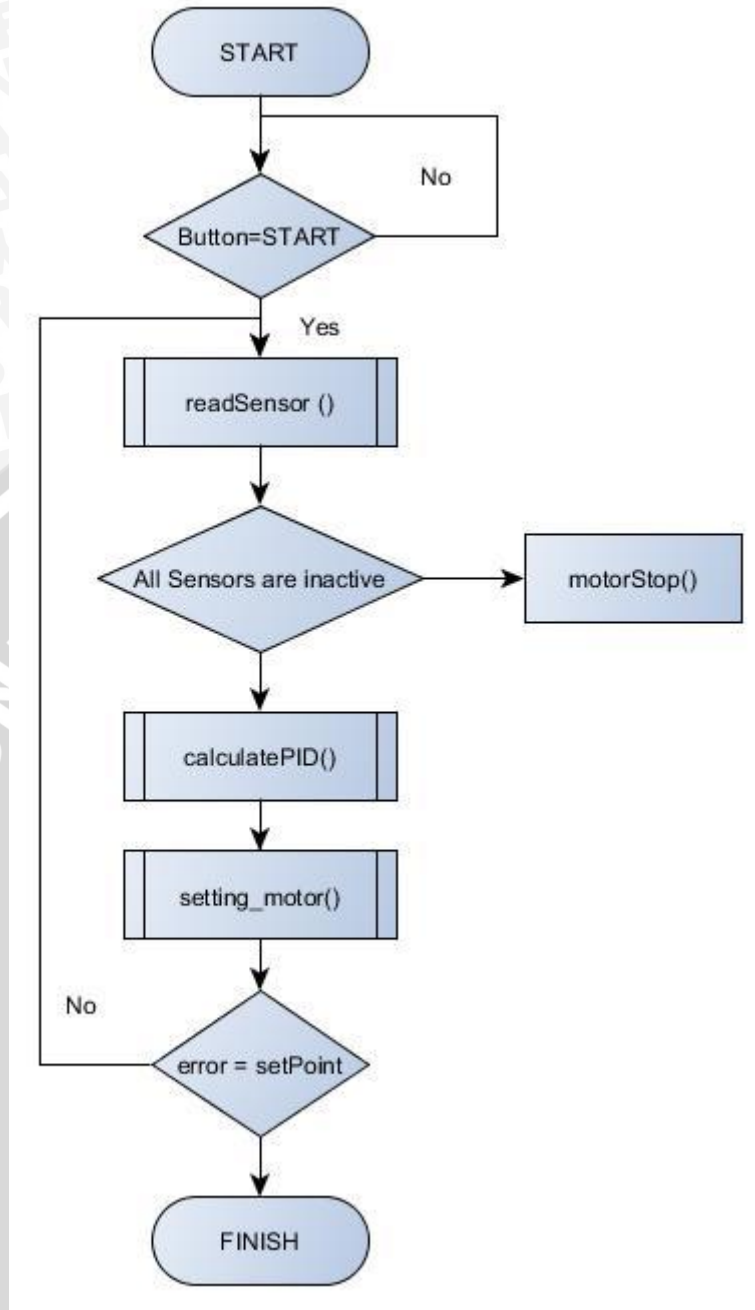


Figure 3.25 Flowchart overall system using PID controller.

From the Figure 3.26 describes flowchart system software using PID controller and RTOS ChibiOS. Flowchart on this almost the same as using controller, but made in memory allocation multitasking with any commands, on reading sensor, compute PID, motor action, and led indicator has its own task. Expected performance software would more quickly.

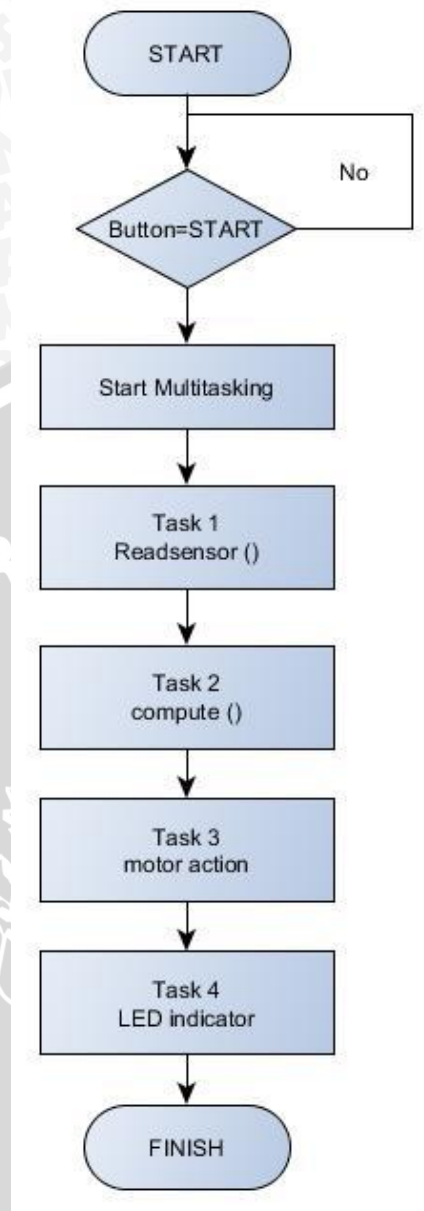


Figure 3.26 Flowchart overall system using PID controller and RTOS ChibiOS.

The distribution of the software design is divided into three parts, namely low-level software, the medium -level software, and high- level software. The section will be described below.

3.5.1 Low-Level Software

The low – level software is a subroutine that interacts directly to the hardware as a driver (Zakaria, 2007). This section describes the motor settings, and LED settings.

3.5.1.1 Mottors Setting

On the motor, program has two differences. Which using the bang- bang controller and the motor program using the PID controller. In the program using the bang-bang controller described in the Figure 3.27 below. In this program changes the speed of the motor using the mapping.

```
void kecil_Forward(uint8_t speed1, uint8_t speed2) {
  speed1=map(speed1,0,100,0,255);
  speed2=map(speed2,0,100,0,255);
  //MOTOR1
  analogWrite(3,0);      //PWM1
  analogWrite(5,speed1); //PWM2
  //MOTOR2
  analogWrite(6,speed2); //PWM3
  analogWrite(9,0);      //PWM4
}
```

Figure 3.27 Programming code using bang-bang controller.

In the program using PID controller. Motors program using PID controller is divided into three, namely the state forward, turn and stop. Where the state of the robot straight road ahead is regulated PWM both motors regulated by 255. Here is Arrow- bot go straight programs.

```
void motor_straight ()
{
  analogWrite (motorkiri, 0);
  analogWrite (motorkiri2, 255);
  analogWrite (motorkanan, 0);
  analogWrite (motorkanan2, 255);
}
```

Figure 3.28 Programming code straight.

The second state is the configuration (motor 1) the right motor PWM varying variables and left motor PWM at 255, and (motor2) otherwise where shows the state turn right and left. The third state is a state in which both motors stop PWM is set at 0.

```

void setting_motor (int motor,int pwm)
{
    if (motor==1){
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, 255);
        analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, pwm);
    }
    if (motor==2){
        analogWrite (motorkiri, 0);
        analogWrite (motorkiri2, pwm);
        analogWrite (motorkanan, 0);
        analogWrite (motorkanan2, 255);
    }
}

```

Figure 3.29 Programming code motor 1 and motor 2.

3.5.1.2 LEDs Setting

LED configuration settings functioned as an indicator. There are 3 LEDs, who has the function of each. On the left LED indicates when the left side of the IR sensor detects the black line, the LED will light, as well as the central LED will light when the IR sensor detects the center of the black, and the right LED will light when the right IR sensor detects the color black.

```

if(reading1 >=30) {IRL_STATUS=HIGH; LEDL_ON;} else {IRL_STATUS=LOW; LEDL_OFF;}
if(reading2 >=30) {IRM_STATUS=HIGH; LEDM_ON;} else {IRM_STATUS=LOW; LEDM_OFF;}
if(reading3 >=30) {IRR_STATUS=HIGH; LEDR_ON;} else {IRR_STATUS=LOW; LEDR_OFF;}

```

Figure 3.30 Programming LED setting.

3.5.1 Medium-Level Software

The medium level of software is here to explain about the IR sensor in determining the outcome. Divided into two algorithm in determining the value of the line and the flowchart in software development.

3.5.1.1 The Line Detection Algorithm Using Bang-Bang Controller

Reflective optical (IR) sensor is due to an analog sensor then each value converted to digital form '0/white line' or '1/black line'. If value of each sensor is less than or equal to 30 the value is declared 0 / detecting white color and the value is more than 30 the value is declared 1 or detecting black color. The three data from sensors are converted into a command for sharp turn left, medium turn left, sharp turn

right, medium turn right, straight and stop. All the commands have been determined speed before in each motor. Arduino code will be explained as follows.

Table 3.13 Microcontroller algorithms for Arrow-bot robot (bang-bang controller).

Left Sensor	Middle Sensor	Right Sensor	PWM(%)
1	0	1	(100,100)
1	1	0	(80,100)
1	0	0	(0,100)
0	1	1	(100,80)
0	0	1	(100,0)
0	0	0	stop

3.5.1.1 The Line Detection Algorithm Using Analog Input

Each combination of optical reflective sensors will give a different product and will determine the direction movement of the robot. Restrictions on the value of the sensor is given on coding because a variety of sensor values when the same color range. So the value of the white rated 30 although the value obtained is less than 30. And for black rated 150 if the value obtained exceeds 150. The determination of this value has been tested with kinds of materials.. Table 3.13 shows the microcontroller algorithms of Arrow-bot robot.

Table 3.14 Microcontroller algorithms for Arrow-bot robot (analog concept).

Left Sensor	Middle Sensor	Right Sensor	Actions of Robot
Range 1 = 30	Range 2 = 30	Range 3 = 30	Stop
Range 1 = 30	Range 2 > 30	Range 3 = 30	Forward
$30 < \text{Range1} \leq 150$	$30 \leq \text{Range1} \leq 150$	Range 3 = 30	Turn Left
Range1 = 30	$30 \leq \text{Range1} \leq 150$	$30 < \text{Range3} \leq 150$	Turn Right

In this method the value of width line as reference of set point. Assume that the coordinate of the middle sensor is 0, and the distance between middle and left sensor, middle and right sensor sensors are considered as 7,5. The output is higher when the sensor is closer to the line. Assume y_1 is the difference between middle and left sensor readings. And is the difference between middle and right sensor readings. Then can find the following relationships between the coordinate of the sensors and the output values:

$$y_1 = ax^2_1 + bx_1 + c, y_2 = ax^2_2 + bx_2 + c \quad (\text{Eq 3.4})$$

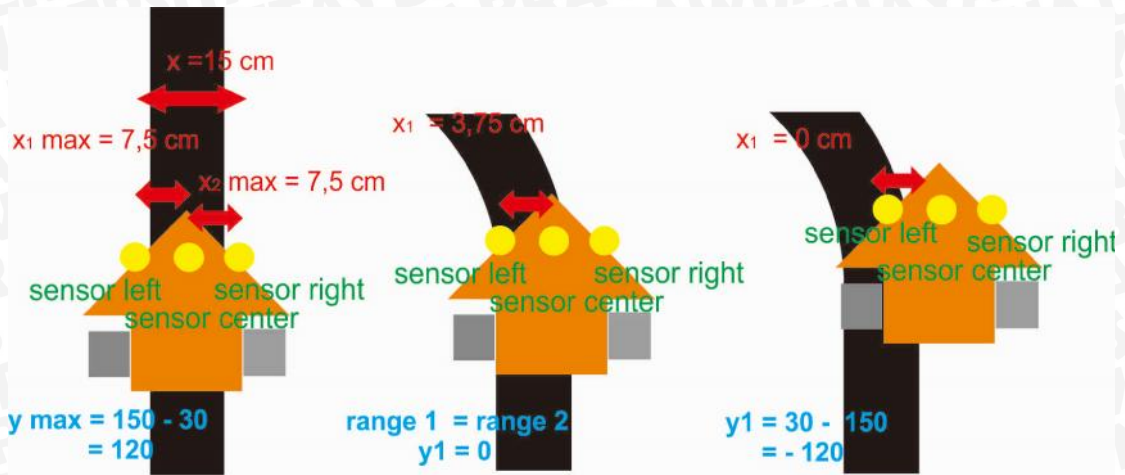


Figure 3.31 Analog sensors reflective position and measurement for algorithm (turn left).

Figure 3.31 shows the comparisons of the true line positions. And the estimation by using (3.4) obtained following calculation.

$$120 = 7,5^2 a + 7,5 b + c \Leftrightarrow 120 = 56,25a + 7,5 b + c \quad (\text{Eq 3.5})$$

$$0 = 3,75^2 a + 3,75 b + c \Leftrightarrow 0 = 14,0625a + 3,75 b + c \quad (\text{Eq 3.6})$$

$$-120 = 0 \cdot a + 0 \cdot b + c \Leftrightarrow 120 = c \quad (\text{Eq 3.7})$$

Substitution obtained from the equation (3.5-3.7)

$$a = 8,53$$

$$b = -47,98$$

$$c = -120$$

So that the equations used in this robot line detection algorithm is $y_1 = 8,53x_1^2 - 47,98x_1 - 120$ and $y_2 = 8,53x_2^2 - 47,98x_2 - 120$, where $y_1 = \text{range 2} - \text{range 1}$ and $y_2 = \text{range 2} - \text{range 3}$. y_1 shows turn left and y_2 shows turn right. x is a variable that is sought after as a determination of error where the set point is set at 7.5. Which explains half of width line 0.75 cm. The roots are used to determine the error. So the value of x as follows.

$$x_{1,1}, x_{1,2} = \frac{47,98 \pm \sqrt{47,98^2 - 4 \cdot 8,53 \cdot (-(120 + y_1))}}{2 \cdot 8,53}, \quad (\text{Eq 3.8})$$

$$x_{2,1}, x_{2,2} = \frac{47,98 \pm \sqrt{47,98^2 - 4 \cdot 8,53 \cdot (-(120 + y_2))}}{2 \cdot 8,53}$$

From two roots values, the value that qualifies as error is positive value. In the Figure 3.32 will explain flowchart how the program works sensor readings and describes the stages in the making of program.

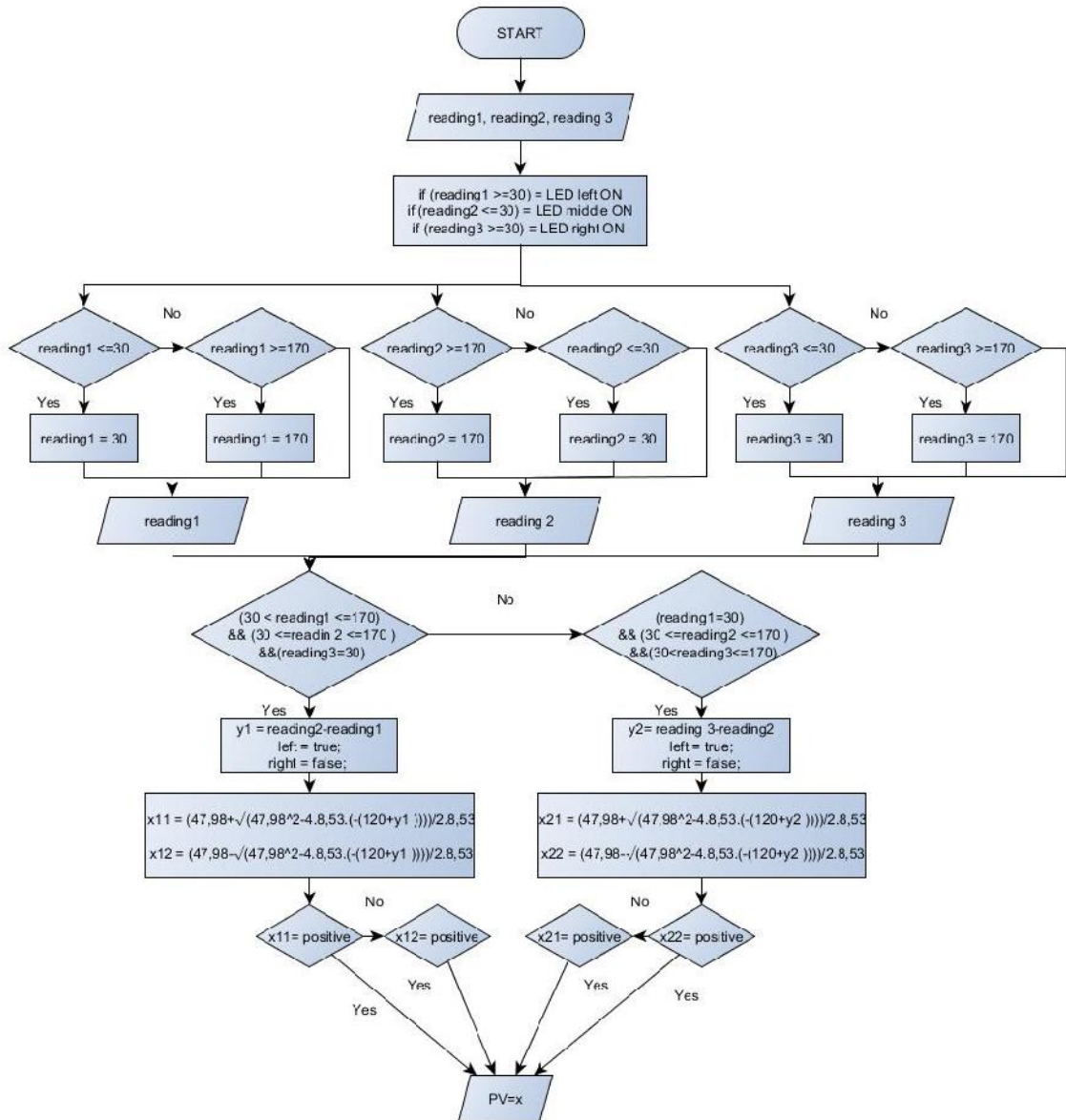


Figure 3.32: Flowchart IR sensors program reading

3.5.1.2 The Line Detection Algorithm Using Digital Input

In this method the three values obtained sensor used as a digital input that is same method with line direction algorithm on bang-bang controller. The following table shows the algorithm of this sensor works. The following table shows the algorithm of this sensor works .

Table 3.15 Microcontroller algorithms for Arrow-bot robot (digital concept).

Left Sensor	Middle Sensor	Right Sensor	PV
1	0	1	0
1	1	0	1
1	0	0	2
0	1	1	-1
0	0	1	-2
0	0	0	stop

After the PV value obtained. Then the value is used to find the value of the error where error = setpoint - PV and setpoint = 0.

3.5.1 High-Level Software

High levels of this software as the final computation in the course of all programs, all of the data from the low-level software, the medium-level software as a reference to the manufacture of high-level software. This section describes the PID calculation and how to control the line follower robot.

3.5.1.1 PID Parameter Calculation

In PID controller that has been described in equation (2.17) is as follows:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (\text{Eq 3.9})$$

From this formula can be formulated into a discrete PID with Transformation Z. Thus the transformation form a discrete PID controller becomes:

$$u(k) = kP \cdot e_k + KIT \sum_0^k e_k + \frac{1}{T} K_D (e_k - e_{k-1}) \quad (\text{Eq 3.10})$$

In the programming language was changed to.

$$\text{PID} = (kP \times \text{error}) + ((KI \times (\text{Error} + \text{last_error}) \times (Ts)) + \left(\frac{(kD \times (\text{error} - \text{last_error}))}{(Ts)} \right)) \quad (\text{Eq 3.11})$$

So it was changed to.

$$\text{PID} = (kP \times \text{error}) + \left((KI \times s\text{Error}) \times \left(\frac{d\text{Time}}{1000} \right) \right) + \left(\frac{(kD \times d\text{Error})}{\left(\frac{d\text{Time}}{1000} \right)} \right) \quad (\text{Eq 3.12})$$

Where :

kP= Proportional Value

Ki= Integral Value

kD = Differential Value

error = serPoint – PV

dTime = (now-Itime)

sError =(sError+error)

$dError = (error - Ierror)$

Flowchart course of the program will be explained in the following section. In the flowchart is divided into two, namely the PID using analog sensors and digital sensors.

3.5.1.2 Flowchart PID Parameter Calculation (Analog Sensor Algorithm)

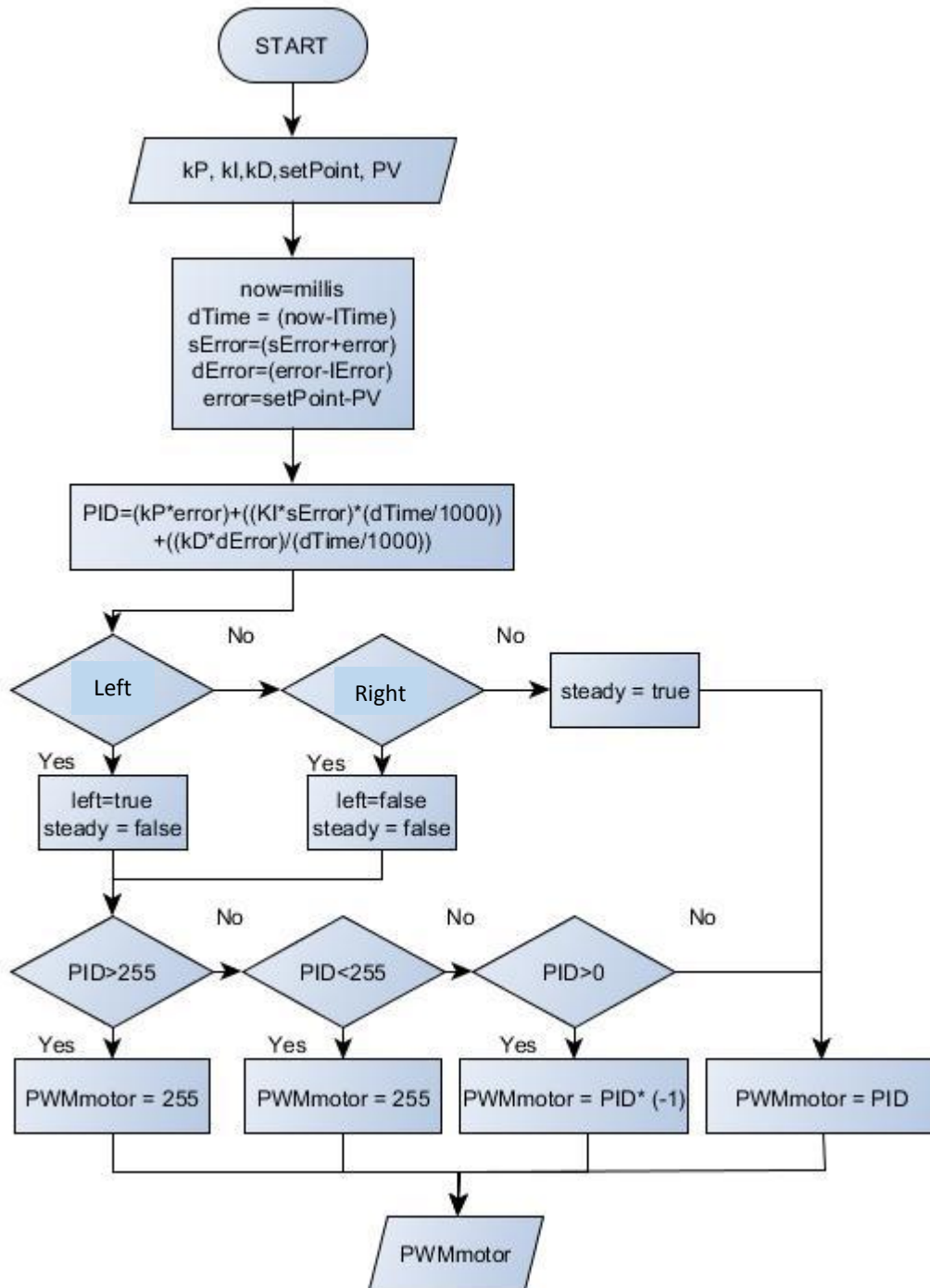


Figure 3.33 Flowchart PID parameter calculation (analog sensor algorithm).

3.5.1.3 Flowchart PID Parameter Calculation (Digital Sensor Algorithm)

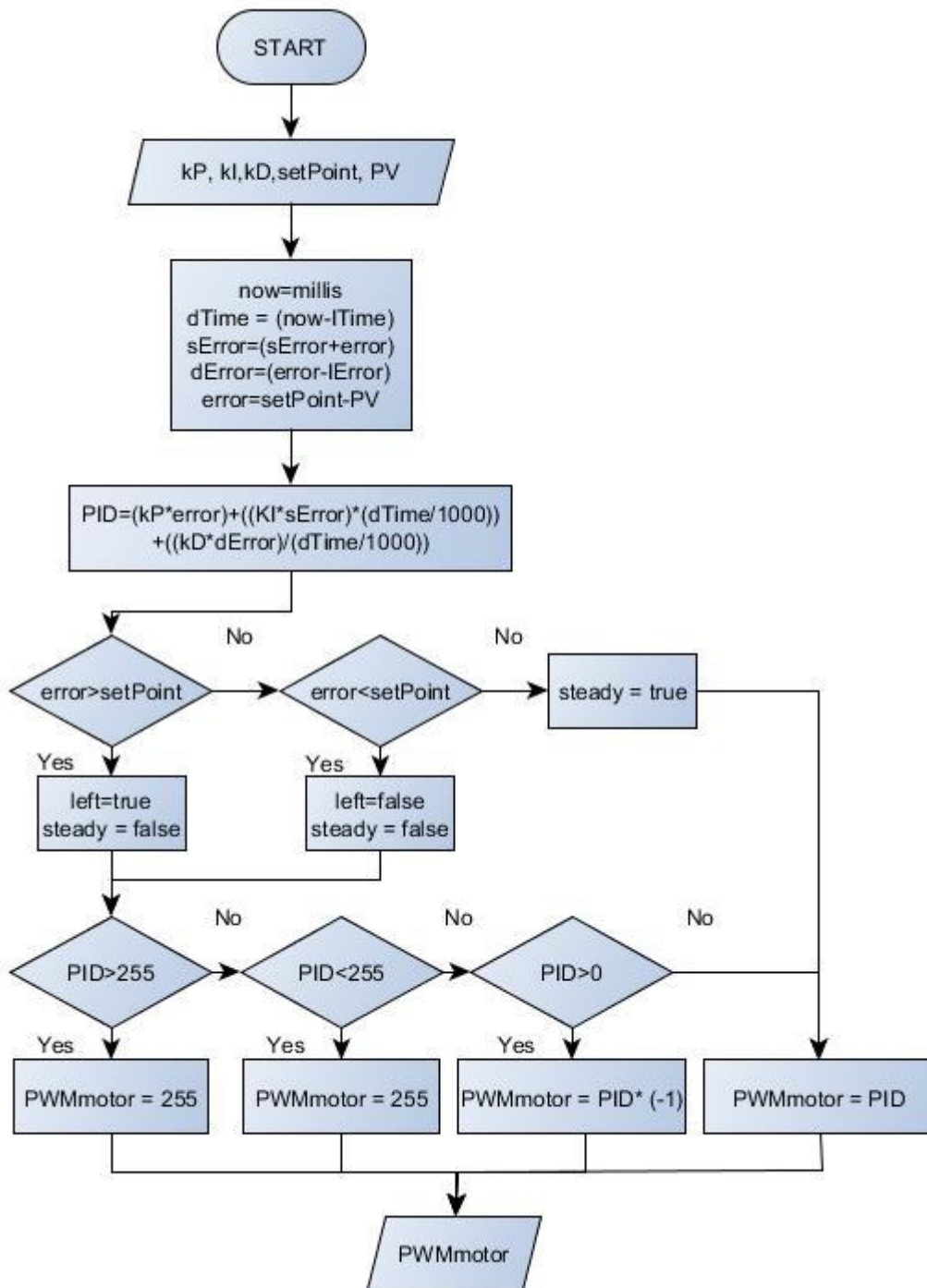


Figure 3.34 Flowchart PID parameter calculation (digital sensor algorithm).