INDIVIDUAL TECHNICAL ANALYSIS

CONTROLLER

Table of Contents

[Introduction 3](#_Toc36387649)

[Controlling Equipment’s 3](#_Toc36387650)

[Conclusion 6](#_Toc36387651)

[References 7](#_Toc36387652)

[APPENDIX – A 8](#_Toc36387653)

# **Introduction**

Purpose of this paper is to do the analytical analysis about the capstone project. And we are working on a project in which a traffic sign cleaner will build that will clean the traffic signs without interrupting the traffic in a usual way. The main objective is to do the cleaning without the interruption and for that purpose we have to design such a mechanism that will send the machine in the air to clean the sign board and control that machine from the controller room. Therefore, we are using the drone with the complete setup of cleaning the sign board and will control that drone through the controller wirelessly from the distance or from the control center.

In this paper I will do the analysis about the controller that will control the drone and will push the water out of the showers. The analysis will help the team in selecting the controller that can control the water in better way and also it will help the team in designing the shower system according to the controller setup.

# **Controlling Equipment’s**

As the controller contains the button which send the signal to the main controller Arduino which will control the water showers. The showers will rotate in different directions and sprinkle the water over the sign board. So starting with the description of controller which has stated below



Figure 1: Remote Controller [1]

This remote controller is basically operating with the radio signal so it is stated as RC (Radio Control) and uses the radio frequency for the transmission of signal. The RC control contains the antenna to transmit the signal and the shower device with the Arduino will receive that signal through the receiver. So the process consists of following two things:

1. Transmitter (Remote)
2. Receiver (Shower)

As the transmitter is the remote which transmit the signal and it takes the signal from the buttons, the buttons have specific code, for example when a forward button will press a specific code will send to the transmitter and that transmitter will send to the receiver. The specific code is basically the frequency, each button has its own frequency but all the buttons contains the frequency in same band, with a small difference in each frequency which can identify at the receiving end easily. The frequency band at which the controller will work is

$$Frequency Band=27 MHz$$

There are six channels present on the remote, by channels means the directional buttons, and all these channels have different frequencies. The channels have described below

1. Forward
2. Forward Left
3. Backward
4. Backward Left
5. Backward Right
6. Forward Right

These channels operate at different frequencies and these frequencies have mentioned below

Table 1: Channels Frequency

|  |  |
| --- | --- |
| **Channel** | **Frequency**  |
| Forward | 27.20 MHz |
| Forward Right | 27.40 MHz |
| Backward | 27.30 MHz |
| Backward Right | 27.60 MHz |
| Backward Left | 27.70 MHz |
| Forward Left  | 27.50 MHz |

As looking at the above frequencies, the channels have difference in their frequencies by 10 MHz or 20 MHz, while the frequency band is same for all the channels. When the button will press, the frequency relate to that button will send to the transmitter and that transmitter will send to its receiver, the receiver will de-code that frequency and send the instruction to the motor to perform that action. As the shower will operate through the motors, so it will rotate the direction of motor and open the shower to sprinkle the water on the sign board.

Consider that the button from the controller is to sprinkle in the forward direction, the button will press, the signal will transmit by the transmitter, the signal will transmit with the frequency of 27.20 MHz and the receiver on the Arduino will receive that frequency, it will recognize the instruction that move the shower in the forward direction and open it. It will send the signal to the motor controller to move the motor towards the front side, as soon the motor will rotate to the front side, the shower will also open and starts sprinkling the water on the sign board.

Another thing to consider here is that the RC controller uses the frequency band of 27MHz, which takes less bandwidth as comparing to the 49 MHz, which takes more bandwidth. From the results it is clear that radio frequency is better to use for controlling the shower system wirelessly. Here is the flowchart describing the complete process of controlling the shower through the remote.



Figure 1: Controller Flowchart

**Motor Driver Controller**

This controller will control the stepper motor to rotate in different directions and move the shower face to shower on the sign board. In this way the process of controlling the shower will control through this driver circuit. This driver circuit will rotate the motors at different angles and stop them there for showering.

**Stepper Motor**

Stepper motor is a dc motor which can rotate at the angles, a motor can either move in 360 degrees in clockwise direction or it can move in anti-clockwise direction as well. So at the input an angle will give to the motor and it rotate to that angle.

The controller of the motor has shown below which will control the motor and rotate the motor to the required position in both clockwise and anti-clockwise direction for the cleaning purpose.



Figure 2: Motor Controller [2]

And the motor controller specifications have given below

Table 2: Motor driver specifications

|  |  |
| --- | --- |
| **Specification** | **Details** |
| Input Voltage | 5V – 36V DC |
| Input Current | 15A |
| Phase  | 3-Phase |

The controller uses with the motor controller is Arduino so Arduino will code to control the motors. Arduino will be the medium between remote and motor controller and it will push the motor driver to move the motor rotor at a specific angle. And the code to control the stepper motor using the Arduino has given in the Appendix A [3].

# **Conclusion**

In this assignment the analysis has done for the controller that will use in the traffic sign cleaning project. The purpose was to analyze the remote control and determine the way to control the water shower so in this analysis a complete process has described through which the controller will control the water shower. In the analysis it has stated that RC remote control will use that will operate on the radio frequency. Arduino will use at the receiving end, which will control the motor drive and then the stepper motor and the stepper motor connects with the shower to control the shower direction. This analysis will help the team in selecting the final design while implementation and it will help the team in selecting the type of showers that will connect with the motors. Furthermore, this analysis will help the team in finalizing the motor controllers.

# **References**

#### [1] H. Ron, “DX6i 6 Channel RC Remote”, available [online], <https://www.horizonhobby.com/dx6i-6-channel-dsmx-reg%3B-transmitter-spmr6630>

#### [2] E. Flow, “Wi-Fi Stepper Motor”, available [online], <https://www.electronics-lab.com/wi-fi-stepper-motor-controller/>

#### [3] S. Arc, “Stepper One Revolution Arduino”, available [online], <https://www.arduino.cc/en/Tutorial/StepperOneRevolution>

# APPENDIX – A

#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution

// for your motor

// initialize the stepper library on pins 8 through 11:

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

int stepCount = 0; // number of steps the motor has taken

void setup() {

 // nothing to do inside the setup

}

void loop() {

 // read the sensor value:

 int sensorReading = analogRead(A0);

 // map it to a range from 0 to 100:

 int motorSpeed = map(sensorReading, 0, 1023, 0, 100);

 // set the motor speed:

 if (motorSpeed > 0) {

 myStepper.setSpeed(motorSpeed);

 // step 1/100 of a revolution:

 myStepper.step(stepsPerRevolution / 100);

 }

}