PaintBot

Final Report

https://github.com/AbuJabal-Hussein/IOT-PaintBot

Submitted by:

Hussein Abu Jabal

Aseel Shehadeh

Rajeh Ayashe

Supervised by:

Yaron Porat

Table of Contents

Abstract

Drawing big pictures in a large scale can be a hard task, especially when the picture is very detailed, since it is hard to see the whole picture at the same time. Likewise, for a construction builder, it takes time and effort to take the right measurements for walls prior to building them in order to build the walls more accurately.
In this paper, we discuss a solution for these similar problems, by proposing a robot which performs these tasks for you, controlled by a mobile application. The suggested robot takes it's input from the mobile application as a miniature sample of the desired picture, and draws the actual picture on a big canvas or on the ground.

1. Introduction

In this paper, we will represent a solution that makes the task of drawing a big-scale picture easier. The suggested solution depends on the modern IOT technologies and cloud database services by proposing the use of a special robot which is controlled by a mobile application.
The user draws whatever he wants using the mobile application, and the robot perform the dirty work of translating the input picture into a big-scale painting on a large canvas or on the ground.
In the following sections, we will discuss the details of implementing and using the product thoroughly.

1.1 General Project Description

As mentioned before, the solution consists of two major parts: a robot and a mobile application. The mobile application provides a simple yet powerful user interface which allows the user to draw whatever he wants smoothly and easily, and allows him to upload the final sketch to the database, in a single click, in order to be processed and drawn by the robot. Moreover, thanks to its implementation technology, it is available on multiple famous platforms including Android, IOS and Web desktop.

The robot is easy to use and can draw any picture that he gets from the mobile application. He listens to the requests that arrive from the mobile application and draws the picture on a canvas or on the ground, as specified by the user. All the user needs to do is paving the floor (for instance, clean it or cover it with paper) and prepare it for drawing.

1.2 Programming Environment

For the implementation of the mobile application, we used the cross-platform SDK of Flutter, which provides us flexibility and support for different mobile platforms including Android, IOS and Web desktop. The development of the app is done using Android Studio IDE.
The robot is provided with an Arduino Uno Board and an ESP8266 WiFi microchip, which are both programmed using Arduino C programming language, using Arduino IDE.
The application and the ESP8266 WiFi microchip communicate with each other using a Firebase Realtime Database.

2. Theoretical Background

2.1 Positioning points in a 2D grid

When it comes to positioning a point on a 2D grid, the most common and simple way to do this is to provide the cartesian coordinates of the point. This is done by specifying a tuple, generally (x, y) where x is the x coordinate or the distance between the projection of the point on the x axis to the origin and y is y coordinate of the point or the distance between the projection of the point on the y axis to the origin. In this way, any complex image or form can be described using a sequence of points, such that when you "join the dots" the image is formed. This is a convenient way of describing the position of a point with respect to an origin.

A point on a 2D grid can also be described using polar coordinates. In this method, the position of a point is described using another tuple, commonly denoted as (theta, r) where theta is the angle between the x axis and the line that connects the origin, and the point and r is the Euclidean distance between the origin and the point.

Figure 1 depicts the transition from polar to cartesian coordinates and the other way transition.

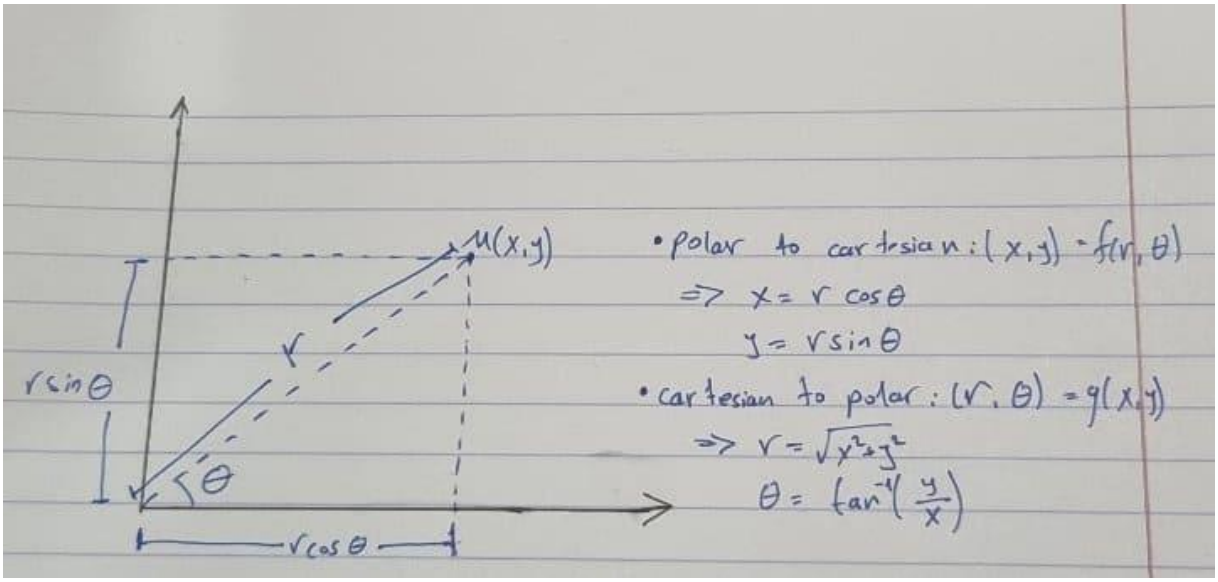


Figure 1

3. Basic System Functionalities

The user can control the system only through the mobile application. The application provides several functionalities and features that make using the robot an easy and smooth task.

3.1 Mobile Application Functionality

The application supports the following features:

- A user can draw on the phone screen using his finger or a special pen.
- A user can draw straight lines.
- A user can draw in a freestyle way.
- A user can erase a single point in a line, or a while line, that he has already drawn.
- A user can clear the screen entirely in a single click.
- A user can undo the last line he drawn in a single click.
- A user can upload the current picture to the cloud in order to be processed and drawn by the robot.

4. Hardware Implementation

4.1 Robot parts

The robot consists of the following parts:

- (2) stepper motors
- (1) servo motor
- (1) Arduino Uno Board
- (1) CNC Shield
- (2) A4988 Drivers
- (1) ESP8266 WiFi Board
- (1) Power supply 12v (to power the CNC shield)
- (2) Power supplies 5v (to power the Arduino Board and the ESP8266 Board)
- (1) micro-USB cable
- (1) USB A-B cable
- (2) wheels
- (1) pen
- (1) matrix
- Several jumper wires
- Wood or kappa boards to build the robot body

Figure 2 shows the structure of the robot:

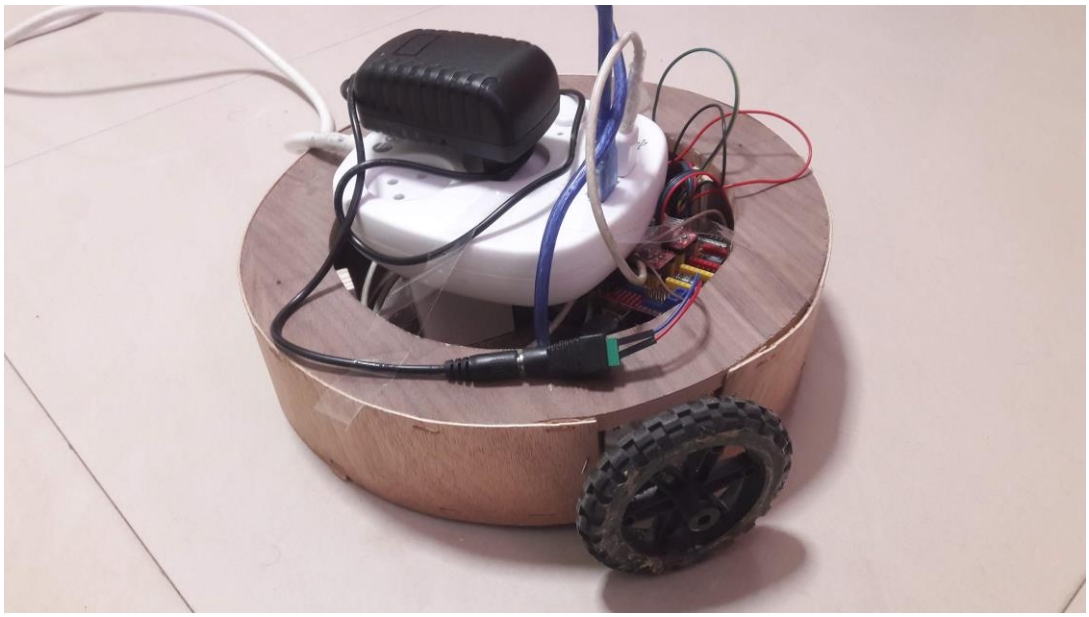


Figure 2

4.2 Building and Wiring

It is recommended to build the robot in round shape, and to place the pen in the middle of the robot (in the origin point of the circle) so it won't have to lift the pen when turning around, so the movement will be easier and more accurate.

Figure 3 shows the circuits and wiring:

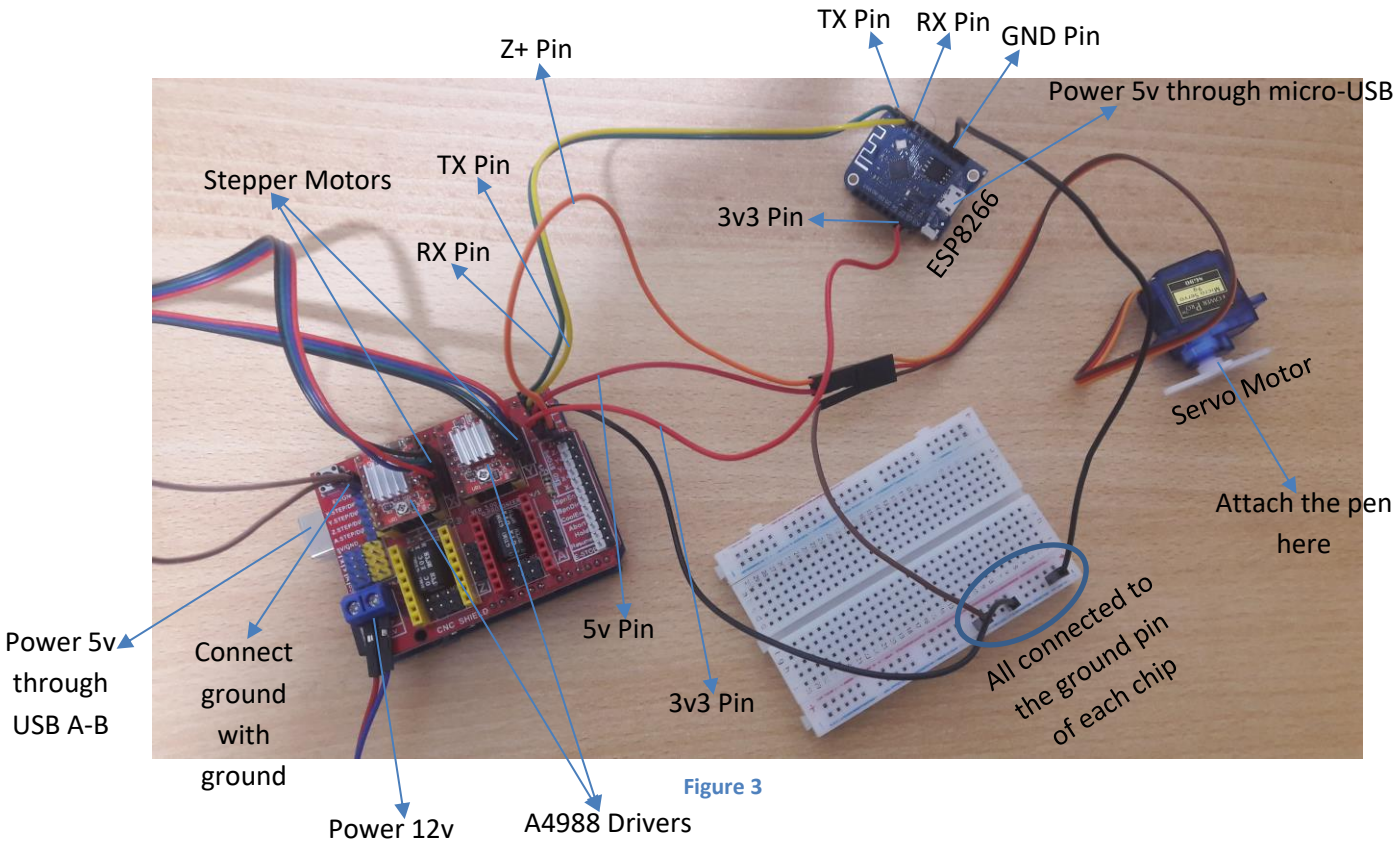


Figure 3

5. Software Implementation

In this section, we will describe the main implementation details of the application.

Definition: A drawing or a picture is a list of paths.

Definition: A path is a list of subsequent (in a matter of the time they were drawn) 2D points.

Note: In order to turn around, the robot moves one stepper motor forward and moves the other one backward simultaneously with respect to the turning angle.

5.1 Modules

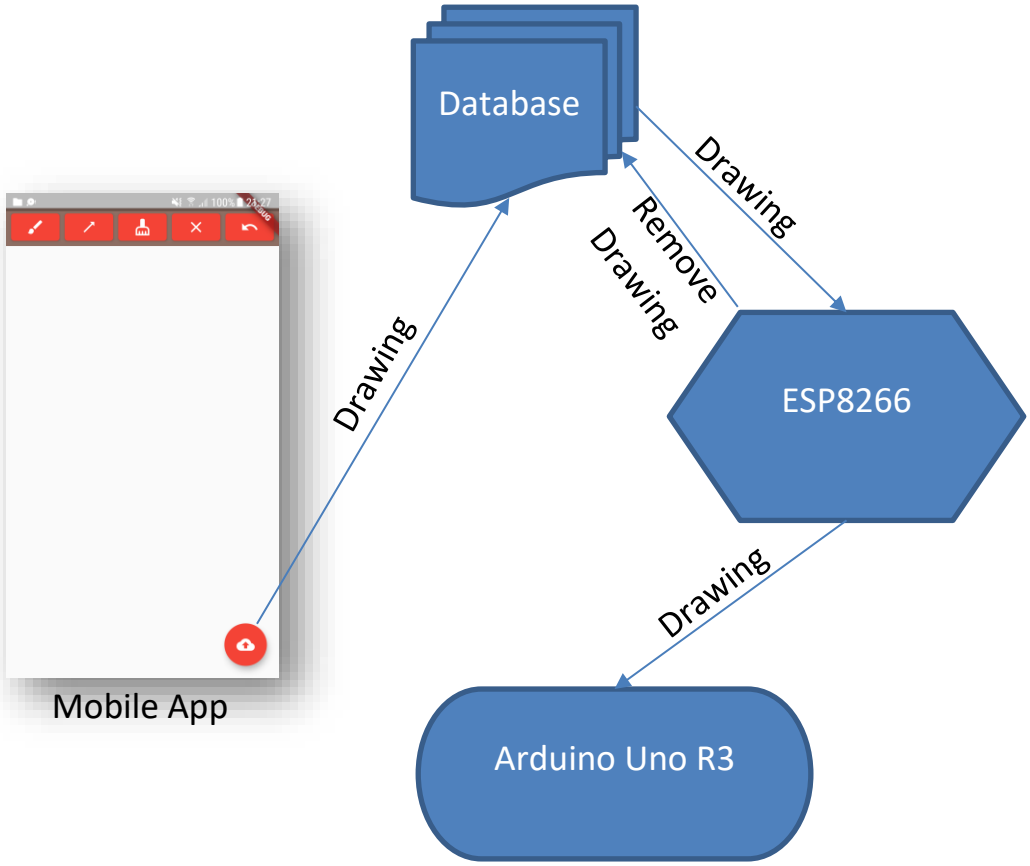
The system is mainly composed of 3 major modules:

- **Mobile Application:**
  This module is responsible for the user interactions with the system, and allows the user to input the drawing of his choice and upload it to the Firebase Realtime Database.

- **ESP8266 WiFi Board code:**
  This module communicates with the Firebase Realtime Database and fetches the drawing, and then sends the drawing to the Arduino Board (using Serial Communication).

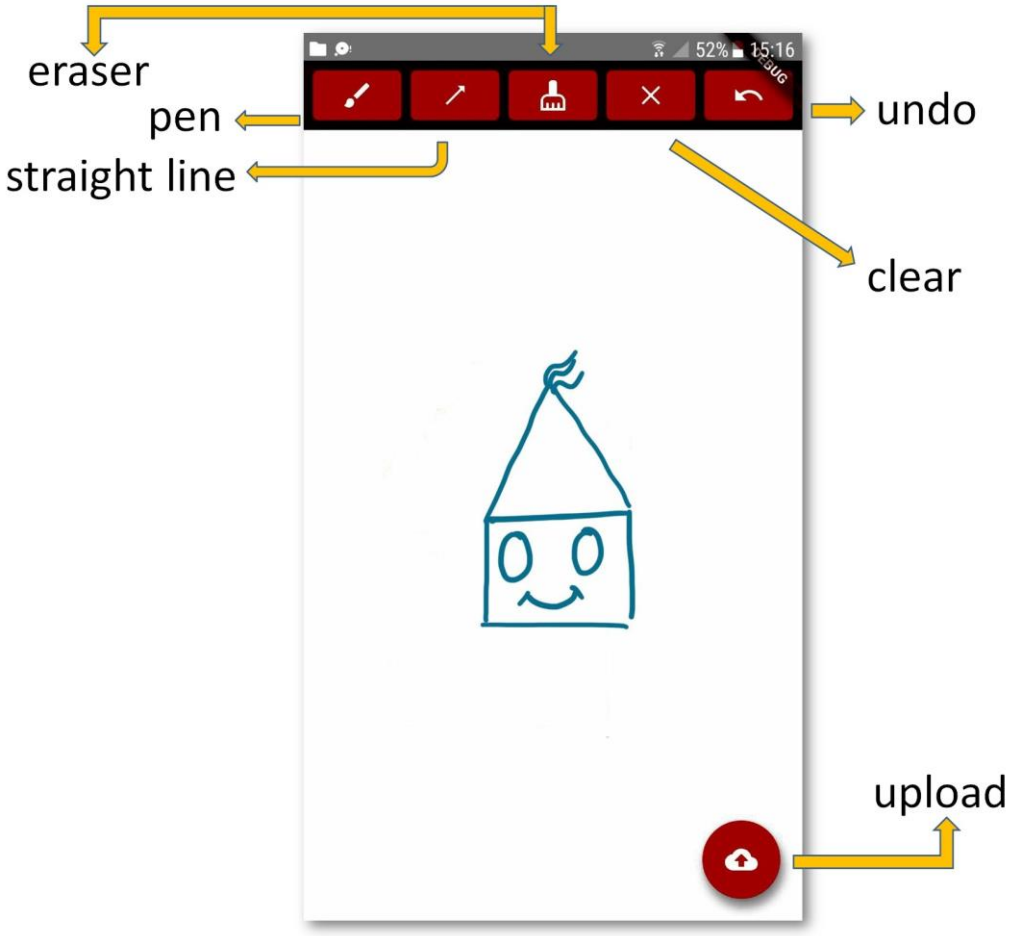
- **Arduino Board code:**
  This module receives a drawing from the ESP8266 board (using Serial Communication), and instructs the motors and the other robot components to behave accordingly after calculating the desired path derived from the drawing and using theories of the cartesian and polar grid representation.

5.2 Top-Level View



5.3 User Interface

The only user interface is through the mobile application:



6. Summary

PaintBot is a technological solution that aims to assist to construction builders determining the locations and shapes of the walls prior to building them according to what described in the building map. Moreover, as a wider goal, it aims to assist to artists drawing pictures on a large scale canvas, and save their precious time.

The PaintBot is controlled via a mobile application, by drawing a certain drawing in the application, and PaintBot takes responsibility to fulfill the destiny of the drawing on a large scale canvas.

7. References

- https://en.wikipedia.org/wiki/Cartesian_coordinate_system

8. Appendix A – User Manual

1. Place the robot on canvas of your choice.
2. Plug the power cable into a power socket.
3. Draw whatever you want using the mobile application.
4. When finished drawing, click the Upload button in the mobile application.

Application manual: