 TEJ-4M1 Culminating Project

Apilash Thiruchelva, Brandon Jagroop Northey, Erfan Alavinia



RC Plane

June 20th 2017

Contents

[Description of Problem Being Solved 3](#_Toc485739817)

[How we fixed the problems 4](#_Toc485739818)

[Team Member Responsibilities 5](#_Toc485739819)

[Schematics and Diagrams 6](#_Toc485739820)

[Receiver Arduino: 6](#_Toc485739821)

[Transmitter Arduino: 6](#_Toc485739822)

[Code for Plane: 7](#_Toc485739823)

[Code for Controller: 9](#_Toc485739824)

[User Manual 11](#_Toc485739825)

# Description of Problem Being Solved

The problem that we are attempting to fix is trying to get a remote controlled plane to fly, by trying to control motors that controls wires running through the inside of the plane controlling the flaps to go up and down and controlling a propeller that would spin at certain speeds. The major problem was getting the something inside the plane in order for us to take control of the motors and activate when necessary. Since it is called a remote controlled plane, we couldn’t just place an Arduino inside the plane and control it from there, so we needed to find something that we could put in the plane and would give us the opportunity to control the plane via a wireless device. So we needed to be able to make a wireless connection from the plane to the controller that linked the Arduino’s and responded to actions made from the Arduino in the controller. From there, we needed to get a controller that would be able to control the planes several actions and hold all of the wiring as well as another Arduino. We needed to take the controller apart and use the board inside it to execute all possible actions. Then, all the wiring in the controller must then be programmed to properly using the Arduino IDE software and would be used to respond to the controller’s actions. But the first physical problem that we had to solve was getting all the wires stripped then soldered properly to the board that we extracted from the controller.

# How we fixed the problems

I’ll start at the very beginning and that’s trying to get the wires soldered to the right joints without making any of them touch. If they happen to touch, the connections wouldn’t be right and when we’d go to test them, they’ll often lead to an outage therefore we had to be careful to make sure the no two joints touched. As mentioned before, we decided to take apart an Xbox 360 controller and use the board of it to control our plane. Since we were in need of three potentiometers, a controller with analogs gave us the best option. We were able to use several inputs on the board of the controller and the rest of the buttons/triggers that we didn’t use, we broke the connection. We then soldered wires to those connections on the board and then tested after every time that we finished to make sure that we weren’t making any mistakes. Since an Arduino was too big to put into the controller, we decided to rip off some of the plastic casing inside the controller, which opened up some space and we were able to stick an Arduino Mini within it. After getting the soldering joints perfect, we searched up a certain way for us to put the wires into the Arduino mini and how to put them into the motors. First we decided to test it wired, straight from the Arduino to the motors on the plane. Once we were able to get all the inputs and programming correct, the propeller began spinning at a speed fast enough to move it off the table. From there, we began to program each input to do a specific function and tested it to see if it would work on the plane.

# Team Member Responsibilities

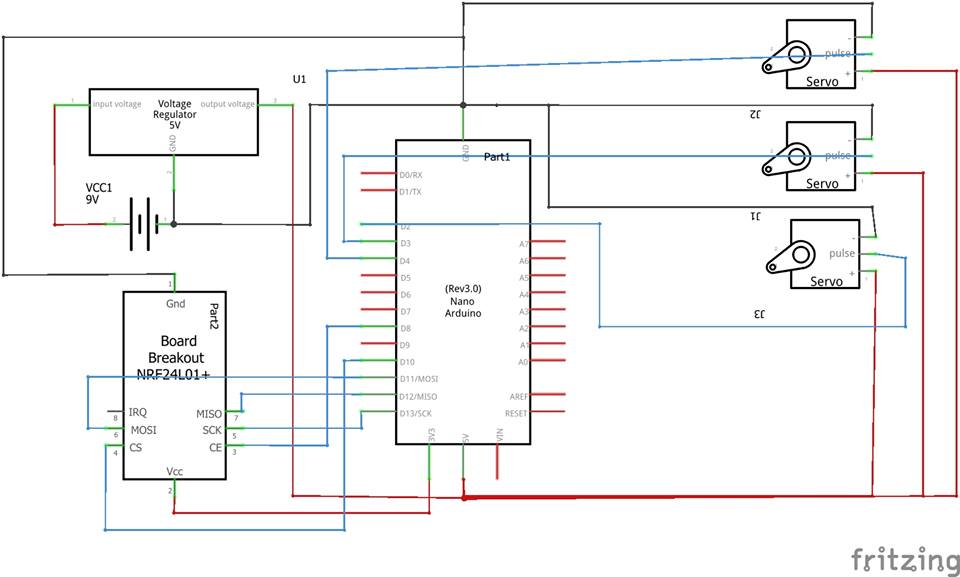
Apilash- was in charge of stripping and soldering wires to the board of the controller and helped in the programming of the plane. Gave ideas of how to fix certain problems and was a key contributor to the wiring of the project.

Brandon- helped with the soldering of wires to the controller’s board and gave some ideas on how to fix certain issues with the project. Created the Project Proposal and attempted to describe all processes of the project and what everyone does.

Erfan- was in charge of programming the Arduino to respond to the commands it was given by the controller and helped in the stripping and soldering of wires to certain joints on the board of the controller. He also helped with the wiring of the controller and and was the main reason for getting the motors to respond to the controller.

# **Schematics and Diagrams**

## Receiver Arduino:



## https://scontent.fybz2-1.fna.fbcdn.net/v/t35.0-12/19369427_1380790252006851_1221869915_o.jpg?oh=99b2db4f14d7a7f23d775078ec5eb42e&oe=594C5723Transmitter Arduino:

# Code for Plane:

//import libraries

#include <nRF24L01.h>

#include <printf.h>

#include <RF24.h>

#include <RF24\_config.h>

#include <Servo.h>

//initialize variables

#define CE\_PIN 8

#define CSN\_PIN 10

#define AELIRON\_PIN 5

#define ELEVATOR\_PIN 6

#define RUDDER\_PIN 9

int throttle;

int aeliron;

int elevator;

int rudder;

//create servo for each part of plane

Servo servo\_aeliron;

Servo servo\_elevator;

Servo servo\_rudder;

//create pipe used for communication between nrf24 devices

const uint64\_t pipe = 0xE8E8F0F0E1LL;

RF24 radio(CE\_PIN,CSN\_PIN);//create radio

int joystick[4];

void setup() {

Serial.begin(9600);

delay(1000);

Serial.println("Receiver Starting");

//start looking for input from transmitting device

radio.begin();

radio.openReadingPipe(1,pipe);

radio.startListening();

//intialize servo motors to specific pin

servo\_aeliron.attach(AELIRON\_PIN);

servo\_elevator.attach(ELEVATOR\_PIN);

servo\_rudder.attach(RUDDER\_PIN);

}//end setup

void loop() {

while(radio.available())

{

//read input

radio.read(joystick,sizeof(joystick));

aeliron = map(joystick[1],0,1023,0,179);

elevator = map(joystick[2],0,1023,0,179);

rudder = map(joystick[3],0,1023,0,179);

//servos in plane for movement.

servo\_aeliron.write(aeliron);

servo\_elevator.write(elevator);

servo\_rudder.write(rudder);

delay(15);

}//end if

}//end main loop

# Code for Controller:

//import libraries

#include <nRF24L01.h>

#include <printf.h>

#include <RF24.h>

#include <RF24\_config.h>

#include <Servo.h>

//initialze variables

#define CE\_PIN 8

#define CSN\_PIN 10

#define JOYSTICK\_X A0

#define JOYSTICK\_Y A1

#define JOYSTICK\_L A2

#define JOYSTICK\_M A3

//create radio

RF24 radio(CE\_PIN, CSN\_PIN);

int joystick[4];

//create pipe used for communication between nrf24

const uint64\_t pipe = 0xE8E8F0F0E1LL; //define transmit pipe

void setup() {

Serial.begin(9600);

Serial.println("RF24 Transmit Starting");

//begin transmitting to specific pipe

radio.begin();

radio.openWritingPipe(pipe);

}//end setup

void loop() {

//read input from analog joysticks

joystick[0] = analogRead(JOYSTICK\_X);

joystick[1] = analogRead(JOYSTICK\_Y);

joystick[2] = analogRead(JOYSTICK\_L);

joystick[3] = analogRead(JOYSTICK\_M);

radio.write(joystick, sizeof(joystick));

}//end main loop

# User Manual

This project is designed to be simple to set up. In order for both the plane and the controller to work, a 9V battery needs to be connected to the Arduino with the transmitter (Arduino in the controller), and a 12V battery is needed in order to power the DC motor inside the plane. Once both are powered, the plane is ready to fly.



Speed throttle

(R2)



Up/Down

Right/Left for when the plane is in the air

Right/Left for when the plane is on ground