FINAL SENIOR PROJECT PRESENTATION: SOLAR POWERED DRONE

Department of Computer and Electrical Engineering and Computer Science

Group 4 Members:
Hernan Aguayo: Computer Engineer (Group Organizer, Software, & Hardware
Enrico Khalil, Computer Engineer (Software & Hardware)
Stephanie Magana, Computer Engineer (Software & Hardware
Jorge Coss, Electrical Engineer (Hardware)
Jose Ortiz, Electrical Engineer (Hardware)
Arturo Valencia, Electrical Engineer (Hardware)

Supervisor: Dr. Wei Li

01 INTRODUCTION

Brief Overview of Project Topic

02 SOFTWARE DEVELOPMENT

Overview of Software Progress

03 HARDWARE DEVELOPMENT

Overview of Hardware Progress

04 INTEGRATION

Overview of Budget, Hardware List, & Integration of Software & Hardware

05 EXPERIMENTAL RESULTS

Goals Met, Results of Project, What we Learned, & Future of our Project

Abstract

This past decade has seen the rise and use of drones for recreational activities and for solving world problems. Current drones are inefficient, rely on electrical energy, and manufactured in ways that harm the environment. Improving current drone tech will result in a greener future for generations to come. Drone tech research can create more interest in STEM for young students of all ages. Interest at an early age will lead to more STEM oriented researchers and graduates. With further research into solar power and testing it may be possible to create drones that are more efficient and greener compared to battery operated drones.

The drone will be powered directly using the sun with the use of solar panels. Without the use of a battery, we can decrease the weight of the drone enabling the use of solar panels. The solar panels will be designed with weight in mind because traditional panels are made of heavy glass. Theoretically the drone should run indefinitely if there are perfect weather conditions. The drone will be controlled via remote control or phone app that the team will design and implement.

Solar powered drones are the future in providing a greener earth even as a small impact. By trading a battery for renewable energy, a drone can be useful for outdoor excursions where the lack of electricity poses a problem to charge a drone. As today's drones consume a lot of energy, solar panel drones will generate a natural source of power/energy, the sun. Not to mention a successfully made solar powered drone can be made open source so that students or young people may be interested in creating and designing drones with inexpensive materials. With the help of our group our solar power drone will be operated via remote control or phone app giving us a wireless solar panel drone. The path to a green future is with greener tech.

01

Introduction

INTRODUCTION

Mass production of commercial drones has led to a widespread popularization of the technology for professional and recreational applications. As [1] states, by widening the application of unmanned aerial drones (UAV) has become a driving factor in developing solar-powered drones in academia and commercial industries. By taking current battery powered drones it is possible to extend the flight time using solar power. However, we wish to eliminate the use of a battery in a drone to power it solely with solar power. By installing solar cells, a solar-powered drone will be capable of drawing solar power to power all the components onboard[1]. By making this combination we hope to manufacture a UAV that is made using environmentally friendly materials and powering it entirely with green energy.

Problem Statement

For the past century drones have become a piece of technology that has had a pervasive impact on our society whether it be for military, industrial, scientific, commercial, and recreational uses. According to the Federal Aviation Administration (FAA), as of August 2021, there have been more than 865,000 drones registered in the United States. Drones can be seen everywhere in our society being used for different purposes and have become affordable for even low-income families. For example, Amazon has seriously considered using drones in delivery in what it is calling the Prime Air service which is said to be able to deliver packages within 30 minutes using unmanned aerial vehicles, also known as, UAV's. It is amazing how current advances in technology allow the use of drones for such purposes, but it raises alarms as to how efficient and environmentally friendly they are. Current drones rely on lithium powered batteries that are 100% recyclable, however lithium mining and processing do have negative impacts on the environment. Not to mention the use of certain plastics for drones require chemical processing that can negatively impact our planet.

We believe that implementing a solar powered drone is a green solution for various industrial, scientific, and commercial applications. Our solar powered drone will completely remove the use of any type of battery so that it is completely powered by the sun. In doing so we are making a UAV that is efficient and environmentally friendly. Such drones will have practical use in only areas with heavy sunlight, yet they will have positive impacts on our society by providing a green solution.

PROJECT TIMETABLE

Delivery Date	Objective Description
Mid November - Mid December 2021	• Continue working on the proposal. • Submission of Final project proposal on December 3rd. • Attending a zoom meeting and presenting Final project proposal on December 3rd.
~~Mid December 2021 – Late January 2022~~ Mid – Late February 2022	• Obtaining all hardware required for our project. • Testing all hardware to ensure it is functioning properly. • Ensuring software on certain hardware is up to date. • Obtained hardware on February 20 & 21, 2022. Received major components later than expected due to shipping delays and relying on third-party sellers for components at a cheaper price.
February 2022 – Early March	• Construction of small-scale prototype using hardware and software. • Performing stress tests on prototype and noting observations. • Observing if software is working properly. • Construction of remote controller prototype. • Expected completion of 25 - 30%. • Current completion as of Feb. 2, 2022, is 20% • We are behind schedule roughly one week
March 2022	• Have final solar drone designs complete. • Begin construction and testing of solar drone model Solaris. • Continue to perform stress tests and fix any bugs. • Expected completion of 50 - 70%.

March 2022	• Have final solar drone designs complete. • Begin construction and testing of solar drone model Solaris. • Continue to perform stress tests and fix any bugs. • Expected completion of 50 - 70%. • As of March 25, we are at 60% completion.
April 2022	• Continue to perform trials and test runs with model Solaris ensuring that the solar drone performs as expected.
Early May 2022	• Achieve 100% completion. • Prepare for Final Project Presentation and Report.

835,000

FAA predicts sharp increase in drone usage by **2023**

SOLAR ENERGY

Solar Energy combines new technology and renewable energy sources. It best meets current environmental, economic, and societal challenges.

HOW THE DRONE WORKS

FEATURES

Solar Powered Drone

FEATURE 01

Lightweight material is used

FEATURE 02

Solar Panel partial implementation

FEATURE 03

Arduino Uno & Nano main hardware

FEATURE 04

Arduino IDE only used for software

02

SOFTWARE

FLIGHT CONTROLLER

Configure:

```
/****************************    The type of multicopter    *******************
  //#define GIMBAL
  //#define BI
  //#define TRI
  //#define QUADP
  #define QUADX
  //#define Y4
  //#define Y6
  //#define HEX6
  //#define HEX6X
  //#define HEX6H  // New Model
  //#define OCTOX8
  //#define OCTOFLATP
  //#define OCTOFLATX
  //#define FLYING_WING
  //#define VTAIL4
  //#define AIRPLANE
  //#define SINGLECOPTER
  //#define DUALCOPTER
  //#define HELI_120_CCPM
  //#define HELI_90_DEG

/****************************    Motor minthrottle    *************************
  /* Set the minimum throttle command sent to the ESC (Electronic Speed Cont
     This is the minimum value that allow motors to run at a idle speed  */
  //#define MINTHROTTLE 1300 // for Turnigy Plush ESCs 10A
  //#define MINTHROTTLE 1120 // for Super Simple ESCs 10A
  //#define MINTHROTTLE 1064 // special ESC (simonk)
  //#define MINTHROTTLE 1050 // for brushed ESCs like ladybird
```

FLIGHT CONTROLLER

Configure:

```
  //#define HELI_120_CCPM
  //#define HELI_90_DEG

/****************************    Motor minthrottle    *******************************/
  /* Set the minimum throttle command sent to the ESC (Electronic Speed Controller)
     This is the minimum value that allow motors to run at a idle speed  */
  //#define MINTHROTTLE 1300 // for Turnigy Plush ESCs 10A
  //#define MINTHROTTLE 1120 // for Super Simple ESCs 10A
  //#define MINTHROTTLE 1064 // special ESC (simonk)
  //#define MINTHROTTLE 1050 // for brushed ESCs like ladybird
  #define MINTHROTTLE 1200 // (*) (**)

/****************************    Motor maxthrottle    *******************************/
  /* this is the maximum value for the ESCs at full power, this value can be increased up to 200
  #define MAXTHROTTLE 2000

/****************************    Mincommand          *******************************/
  /* this is the value for the ESCs when they are not armed
     in some cases, this value must be lowered down to 900 for some specific ESCs, otherwise the
  #define MINCOMMAND  1000

/**********************************    I2C speed   ************************************/
  #define I2C_SPEED 100000L     //100kHz normal mode, this value must be used for a genuine WMP
  //#define I2C_SPEED 400000L   //400kHz fast mode, it works only with some WMP clones

/***************************    Internal i2c Pullups    ********************************/
  /* enable internal I2C pull ups (in most cases it is better to use external pullups) */
  //#define INTERNAL_I2C_PULLUPS

/**************************************************************************************/
/*****************          boards and sensor definitions            ******************/
/**************************************************************************************/

  /****************************    Combined IMU Boards    ********************************/
    /* if you use a specific sensor board:
```

FLIGHT CONTROLLER

Configure:

```
/****************************    Combined IMU Boards    ********************************/
  /* if you use a specific sensor board:
     please submit any correction to this list.
       Note from Alex: I only own some boards, for other boards, I'm not sure, the info was gathe
  //#define FFIMUv1         // first 9DOF+baro board from Jussi, with HMC5843                   <-
  //#define FFIMUv2         // second version of 9DOF+baro board from Jussi, with HMC5883       <-
  //#define FREEIMUv1       // v0.1 & v0.2 & v0.3 version of 9DOF board from Fabio
  //#define FREEIMUv03      // FreeIMU v0.3 and v0.3.1
  //#define FREEIMUv035     // FreeIMU v0.3.5 no baro
  //#define FREEIMUv035_MS  // FreeIMU v0.3.5_MS                                                <-
  //#define FREEIMUv035_BMP // FreeIMU v0.3.5_BMP
  //#define FREEIMUv04      // FreeIMU v0.4 with MPU6050, HMC5883L, MS561101BA                  <-
  //#define FREEIMUv043     // same as FREEIMUv04 with final MPU6050 (with the right ACC scale)
  #define NANOWII       // the smallest multiwii FC based on MPU6050 + pro micro based proc <-
  //#define PIPO            // 9DOF board from erazz
  //#define QUADRINO        // full FC board 9DOF+baro board from witespy  with BMP085 baro     <-
  //#define QUADRINO_ZOOM   // full FC board 9DOF+baro board from witespy  second edition
  //#define QUADRINO_ZOOM_MS// full FC board 9DOF+baro board from witespy  second edition       <-
  //#define ALLINONE        // full FC board or standalone 9DOF+baro board from CSG_EU
  //#define AEROQUADSHIELDv2
  //#define ATAVRSBIN1      // Atmel 9DOF (Contribution by EOSBandi). requires 3.3V power.
  //#define SIRIUS          // Sirius Navigator IMU                                             <-
  //#define SIRIUSGPS       // Sirius Navigator IMU  using external MAG on GPS board            <-
  //#define SIRIUS600       // Sirius Navigator IMU  using the WMP for the gyro
  //#define SIRIUS_AIR      // Sirius Navigator IMU 6050 32U4 from MultiWiiCopter.com           <-
  //#define SIRIUS_AIR_GPS  // Sirius Navigator IMU 6050 32U4 from MultiWiiCopter.com with GPS/MA
  //#define SIRIUS_MEGAv5_OSD //  Paris_Sirius™ ITG3050,BMA280,MS5611,HMC5883,uBlox  http://www.M
  //#define MINIWII         // Jussi's MiniWii Flight Controller                                <-
  //#define MICROWII        // MicroWii 10DOF with ATmega32u4, MPU6050, HMC5883L, MS561101BA from
  //#define CITRUSv2_1      // CITRUS from qcrc.ca
  //#define CHERRY6DOFv1_0
  //#define DROTEK_10DOF    // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
  //#define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
```

FLIGHT CONTROLLER

```
*****************************************************************************************/

/*****************************    PPM Sum Reciver    *************************************/
  /* The following lines apply only for specific receiver with only one PPM sum signal, on digital PIN 2
     Select the right line depending on your radio brand. Feel free to modify the order in your PPM order is different */
  //#define SERIAL_SUM_PPM         PITCH,YAW,THROTTLE,ROLL,AUX1,AUX2,AUX3,AUX4,8,9,10,11 //For Graupner/Spektrum
  //#define SERIAL_SUM_PPM         ROLL,PITCH,THROTTLE,YAW,AUX1,AUX2,AUX3,AUX4,8,9,10,11 //For Robe/Hitec/Futaba
  //#define SERIAL_SUM_PPM         ROLL,PITCH,YAW,THROTTLE,AUX1,AUX2,AUX3,AUX4,8,9,10,11 //For Multiplex
  //#define SERIAL_SUM_PPM         PITCH,ROLL,THROTTLE,YAW,AUX1,AUX2,AUX3,AUX4,8,9,10,11 //For some Hitec/Sanwa/Others
#define SERIAL_SUM_PPM         THROTTLE,YAW,PITCH,ROLL,AUX1,AUX2,AUX3,AUX4,8,9,10,11 // BoykaCopter
  // Uncommenting following line allow to connect PPM_SUM receiver to standard THROTTLE PIN on MEGA boards (eg. A8 in CRIUS AIO)
  //#define PPM_ON_THROTTLE

/***********************    Spektrum Satellite Reciver    *********************************/
  /* The following lines apply only for Spektrum Satellite Receiver
     Spektrum Satellites are 3V devices.  DO NOT connect to 5V!
     For MEGA boards, attach sat grey wire to RX1, pin 19. Sat black wire to ground. Sat orange wire to Mega board's 3.3V (or any other
     For PROMINI, attach sat grey to RX0.  Attach sat black to ground. */
  //#define SPEKTRUM 1024
  //#define SPEKTRUM 2048
  //#define SPEK_SERIAL_PORT 1    // Forced to 0 on Pro Mini and single serial boards; Set to your choice of 0, 1, or 2 on any Mega bas
  //**************************
  // Defines that allow a "Bind" of a Spektrum or Compatible Remote Receiver (aka Satellite) via Configuration GUI.
  //   Bind mode will be same as declared above, if your TX is capable.
  //   Ground, Power, and Signal must come from three adjacent pins.
  //   By default, these are Ground=4, Power=5, Signal=6.  These pins are in a row on most MultiWii shield boards. Pins can be override
  //   Normally use 3.3V regulator is needed on the power pin!!  If your satellite hangs during bind (blinks, but won't complete bind w
  //**************************
  //   For Pro Mini, the connector for the Satellite that resides on the FTDI can be unplugged and moved to these three adjacent pins.
  //#define SPEK_BIND             //Un-Comment for Spektrum Satellie Bind Support.  Code is ~420 bytes smaller without it.
  //#define SPEK_BIND_GROUND 4
  //#define SPEK_BIND_POWER  5
  //#define SPEK_BIND_DATA   6
```

DRONE

Calibration:

ESC_calibrate

```
/*ESC calibration sketch; author: ELECTRONOOBS */
#include <Servo.h>

#define MAX_SIGNAL 2000
#define MIN_SIGNAL 1000
#define MOTOR_PIN1 9
#define MOTOR_PIN2 8
#define MOTOR_PIN3 7
#define MOTOR_PIN4 6

int DELAY = 1000;

Servo motor1;
Servo motor2;
Servo motor3;
Servo motor4;

void setup() {
  Serial.begin(9600);
  delay(1500);
  Serial.println("Program begin...");
  delay(1000);
  Serial.println("This program will start the ESC.");

  motor1.attach(MOTOR_PIN1);
  motor2.attach(MOTOR_PIN2);
  motor3.attach(MOTOR_PIN3);
  motor4.attach(MOTOR_PIN4);


  Serial.print("Now writing maximum output: (");Serial.print(MAX_SIGNAL);Serial.print(" us in this
  Serial.println("Turn on power source, then wait 2 seconds and press Enter");
  motor1.writeMicroseconds(MAX_SIGNAL);
  motor2.writeMicroseconds(MAX_SIGNAL);
  motor3.writeMicroseconds(MAX_SIGNAL);
```

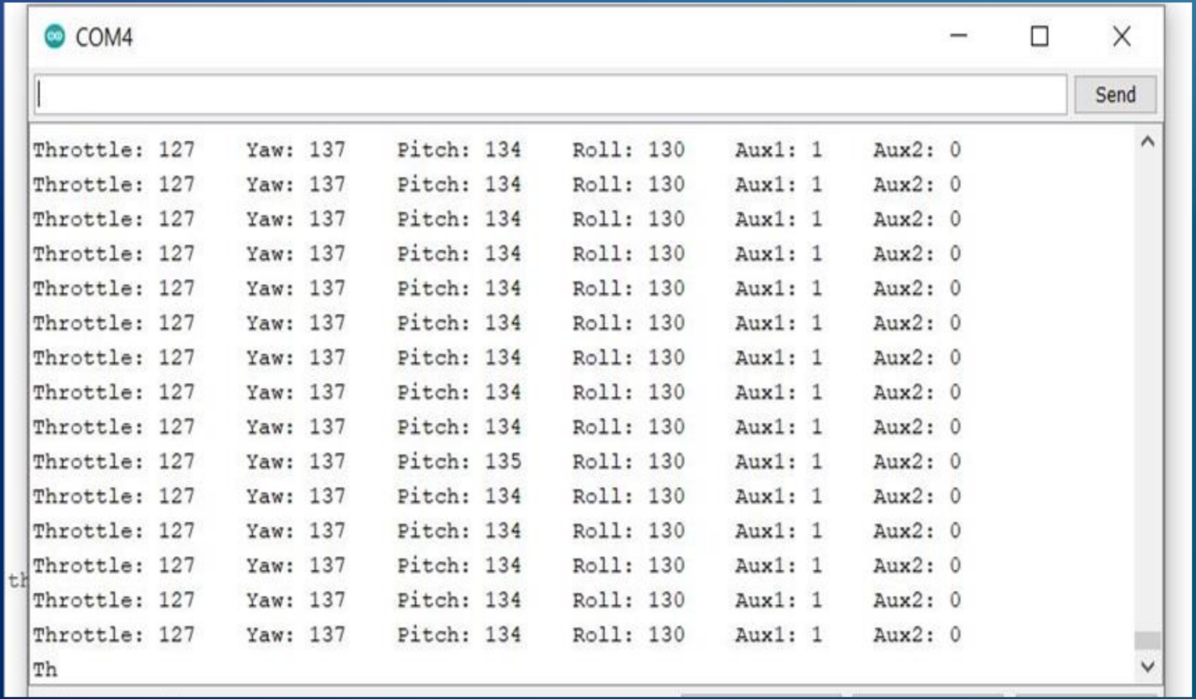
CONTROLLER

Transmitter:

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

/*Create a unique pipe out. The receiver has to
  wear the same unique code*/

const uint64_t pipeOut = 0xE8E8F0F0E1LL; //IMPORTANT: The same as in the receiver

RF24 radio(9, 10); // select  CSN  pin

// The sizeof this struct should not exceed 32 bytes
// This gives us up to 32 8 bits channals
struct MyData {
  byte throttle;
  byte yaw;
  byte pitch;
  byte roll;
  byte AUX1;
  byte AUX2;
};

MyData data;

void resetData()
{
  //This are the start values of each channal
  // Throttle is 0 in order to stop the motors
  //127 is the middle value of the 10ADC.

  data.throttle = 0;
  data.yaw = 127;
  data.pitch = 127;
  data.roll = 127;
```

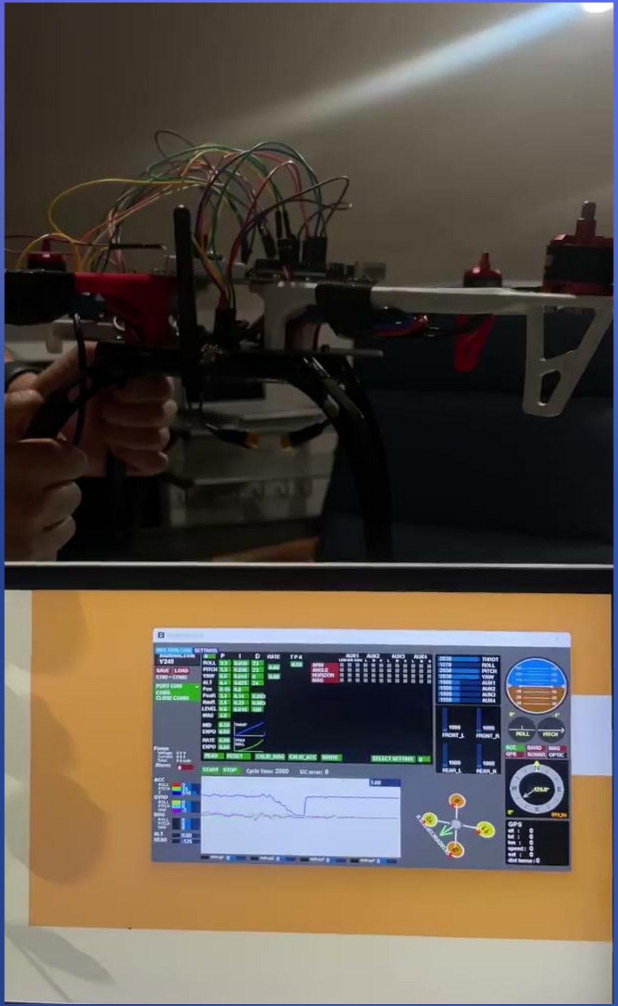
CONTROLLER

Receiver Test:



MULTIWII

Open-source software with GUI application:



03

HARDWARE

HARDWARE DEVELOPMENT

3D PRINTS

Drone casing
AutoCAD

TRANSMITTER & Receiver

NRF24L01 Wireless Transceiver

Motors

Brushless Motor
30 A Electronic Speed Controller

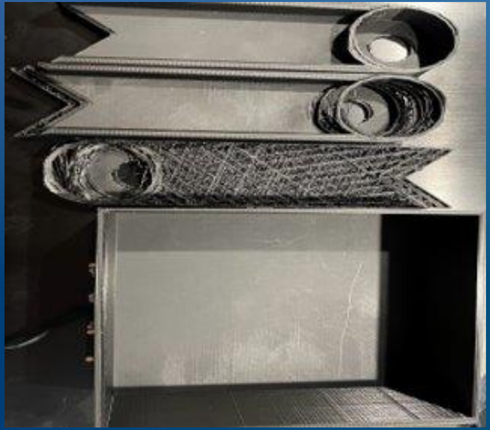
FLIGHT CONTROLLER

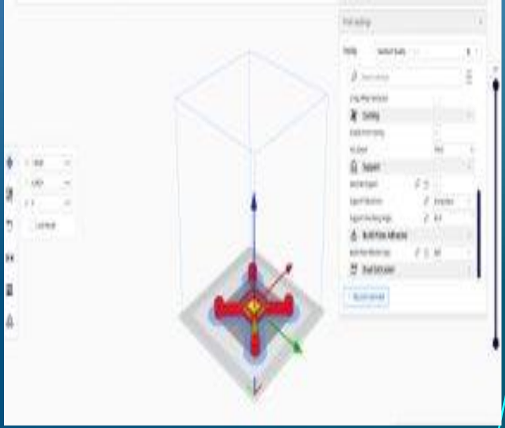
Arduino Nano
3 Axis Gyroscope and Accelerometer sensor

3D PRINTS

HARDWARE:

- 3 different drone Prototypes were created.
- Creating designs on AutoCAD and adjustments on Cura.
- Final rough draft

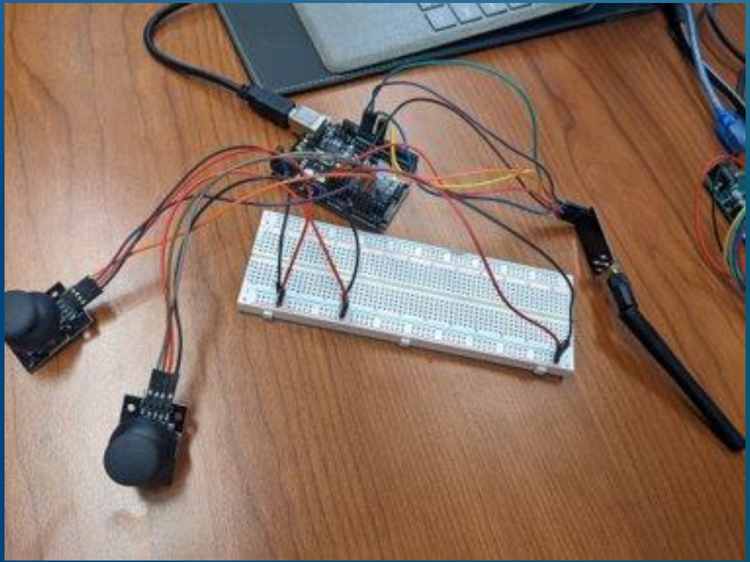




CONTROL TRANSMITTER & CONTROL RECEIVER

HARDWARE:

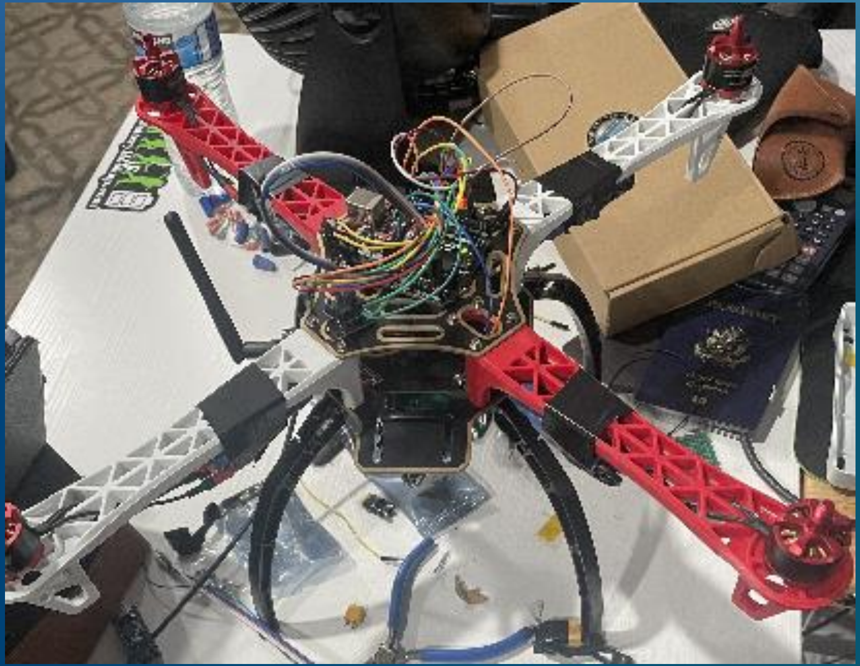
- We used an NRF24L01 module which is a wireless transceiver that can send data at 2.4Ghz
- 100 - 200 ft of range
- Using analog sticks, jumper wires into the ELEGOO Uno



DRONE ACCESSORIES

HARDWARE:

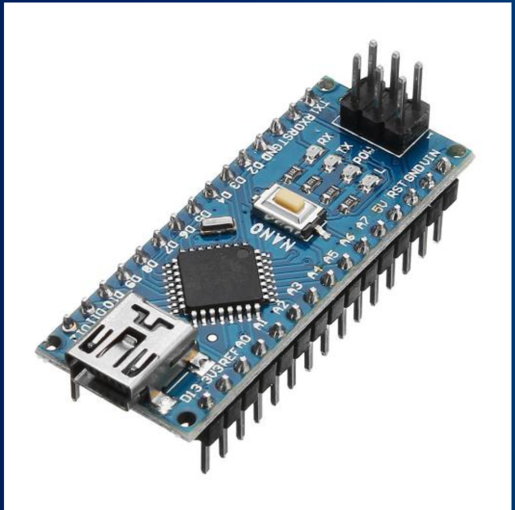
- PCB Board
- Solar Panel
- Arduino Nano
- 3 axis Gyroscope
- Accelerator Sensors
- Brushless motors
- 30A Electronic Speed Controller
- Lipo Battery

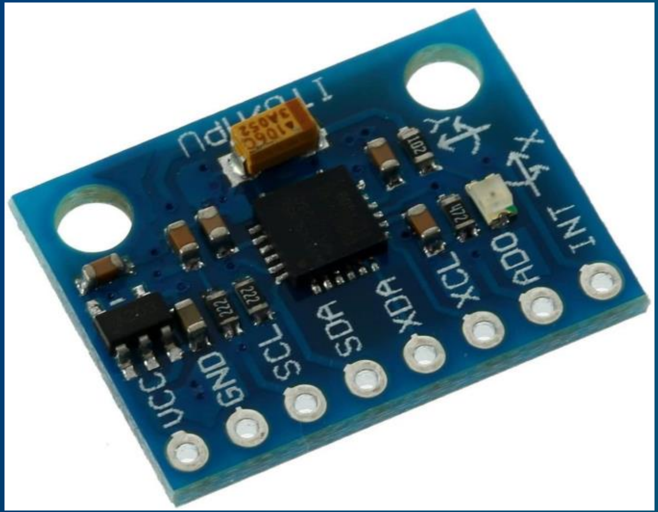


FLIGHT CONTROLLER

HARDWARE:

- Arduino Nano connected to 3 Axis Gyroscope and Accelerometer
- Sensors were programmed to operate the flight controller





04

INTEGRATION

Culmination of Software and Hardware Progress

PROJECT COST, BUDGET, & PARTS LIST

# of PARTS	PARTS DESCRIPTION	COST
2	ELEGOO Uno Board	$80
1	Jumper Wires(30pcs) & Electrical Wires(2ft)	$15
10	Solar Panels(5V, 30mA, 53x30mm)	$15
10	Solar Panels(1V, 80mA, 30x25mm)	$15
1	3D Printing Material	$15

PROJECT COST, BUDGET, & PARTS LIST

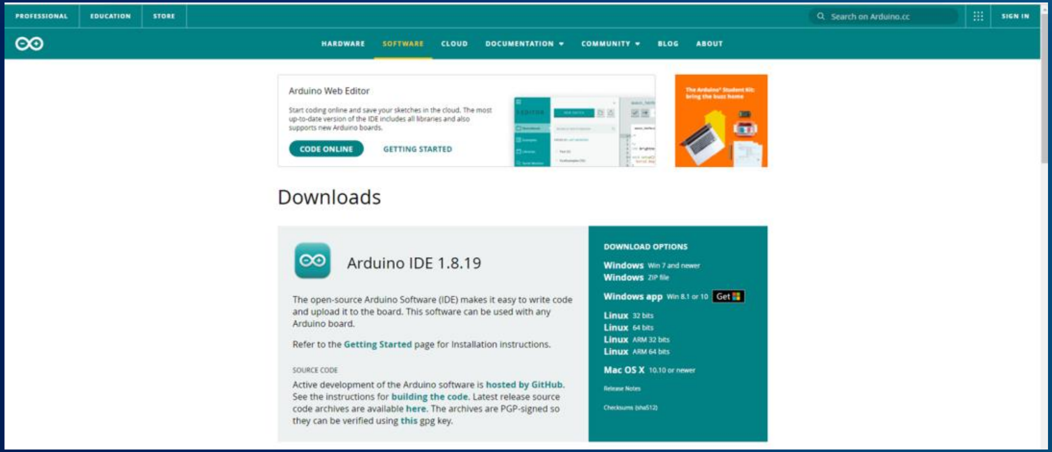
# of PARTS	PARTS DESCRIPTION	COST
4	Joysticks(For Remote)	$13.99
1	WayinTop 2-Set Wireless Transceiver & Receiver	$20
1	HiLetGo 3pcs GY-521 MPU-6050 3-Axis Accelerometer Gyroscope Module	$10
6	ELEGOO Nano Board CH340/ATmega+ 328P	$25
6	Readytosky 10x4.5 Rotor Blades	$20

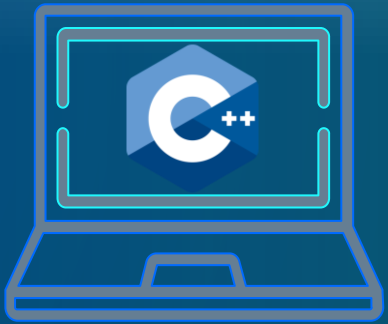
PROJECT COST, BUDGET, & PARTS LIST

# of PARTS	PARTS DESCRIPTION	COST
4	30A RC Brushless Motor Electric Speed Controller ESC	$70
4	ReadytoSky 2212 920KV Brushless Motors CW CCW	$40
4	YoungRC F450 Drone Frame Kit 4-Axis Airframe 450mm for Quadcopter	$22
	Average Price for Parts	$366
	Average Cost Per Group Member	$61

SOFTWARE INTEGRATION

- Arduino IDE(C++) only used for programming
- Flight Controller programmed and calibrated using open-source MultiWii graphical interface
- Drone Remote Controller(Transmitter) programmed and tested
- Drone Receiver programmed and tested





HARDWARE INTEGRATION

- Creation of Flight Controller
- Brushless Motors & Electronic Speed Controllers
- Soldering to PCB
- Sensor & Ground Wires
- NRF24L01 Modules for Receiver & Transmitter
- Partial Integration of Solar Panels
- Necessary to Include LiPo Battery
- 3D Printing Drone Housing

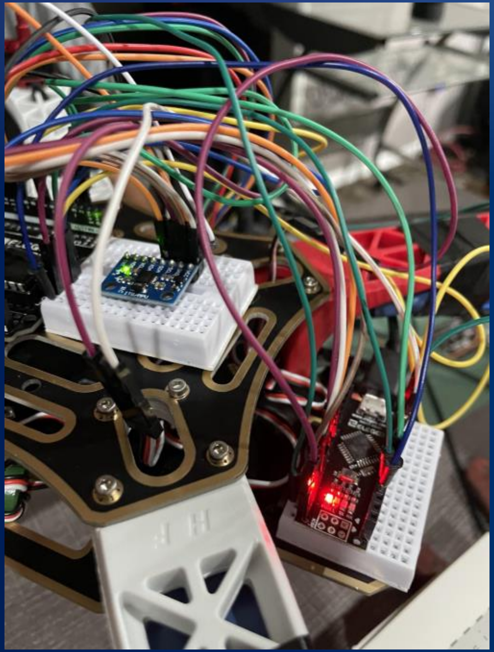
05

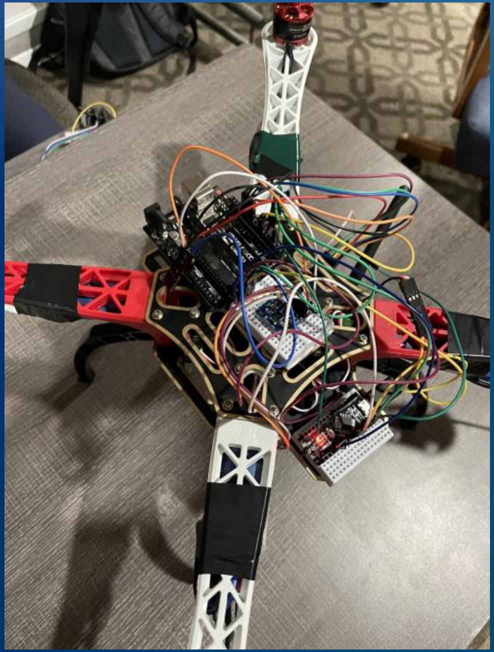
EXPERIMENTAL RESULTS

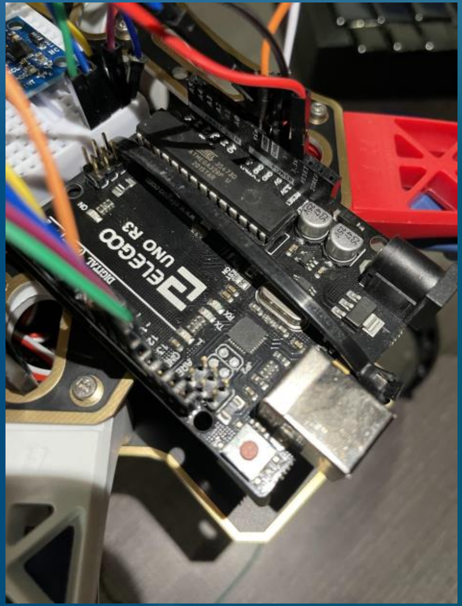
GOALS EXPECTED VS MET

- Goals expected based on our project proposal
  - 100% completion of project
- Goals actually met based on our project proposal
  - Technically did achieve 100% completion
- Goals expected of hardware portion
  - Use of low-cost components
  - Use of 3D printing
  - Use of solar panels
- Goals actually met of hardware portion
  - Did use low-cost components
  - Expensive components were used
  - Did use 3d printing
  - Solar panels were used
- Goals expected of software portion
  - Sole use of Arduino IDE
  - Coding only with C++
- Goals actually met of software portion
  - Only used Arduino IDE
  - Used only C++

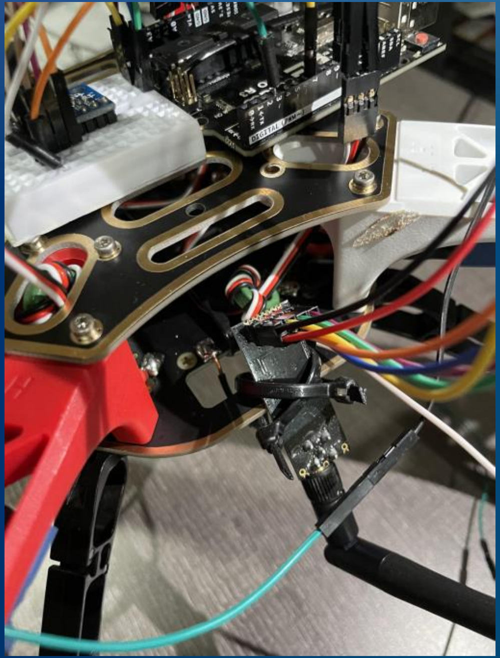
CURRENT COMPLETION OF SOLAR POWER DRONE

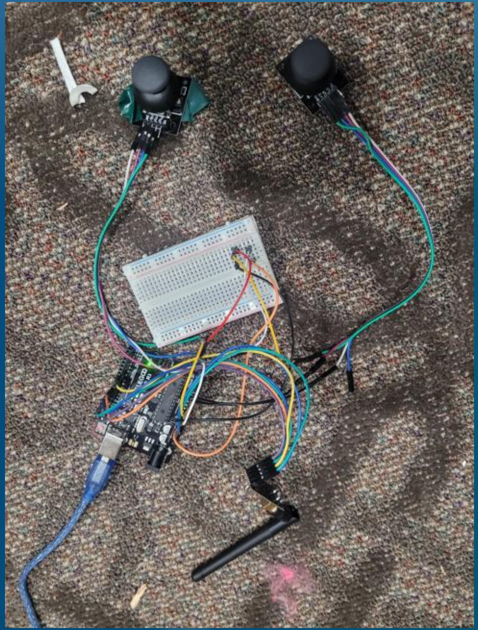






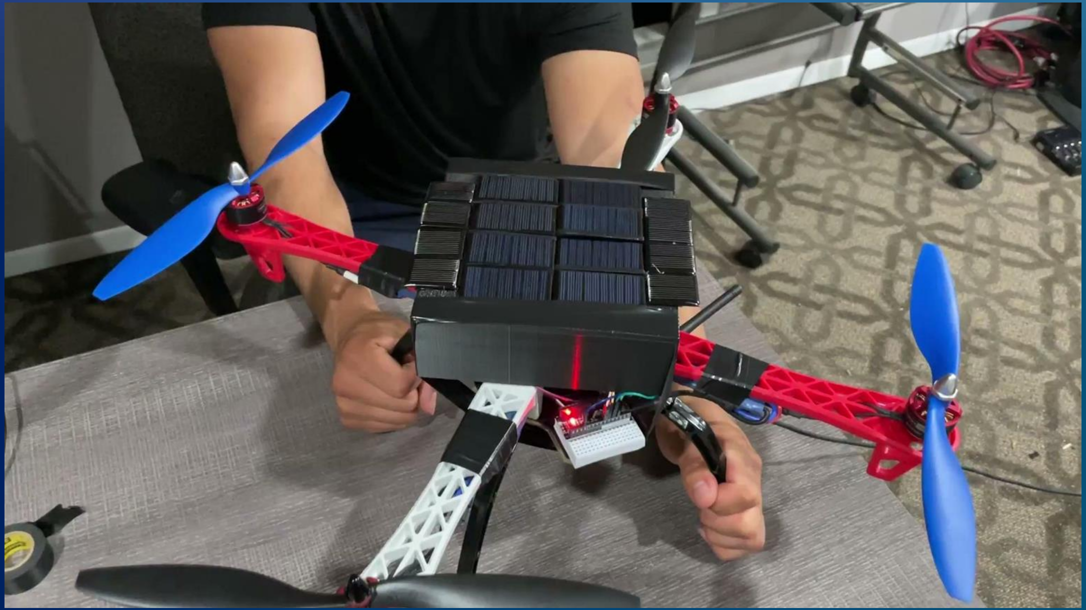
CURRENT COMPLETION OF SOLAR POWER DRONE CONTINUED





CURRENT COMPLETION OF SOLAR POWER DRONE CONTINUED

CURRENT COMPLETION OF SOLAR POWER DRONE CONTINUED

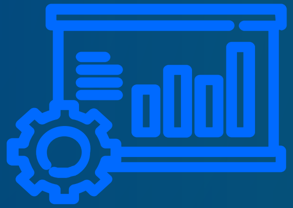


CURRENT COMPLETION OF SOLAR POWER DRONE CONTINUED

CURRENT COMPLETION OF SOLAR POWER DRONE CONTINUED

WHAT WE LEARNED FROM THIS PROJECT

- Teamwork
- Researching
- Problem-Solving
- 3D-Modeling & Printing
- Solar Panels
- Arduinos
- PCB Designing (Fritzing)
- Arduino IDE
- Soldering



FUTURE OF THE PROJECT

- Continuing to work on project
- Upgrading hardware components
- Improving code and software used
- Properly implementing a solar panel
- Ensuring a green running drone
- Designing sleek & aerodynamic 3D model
- Printing a successful model

ACKNOWLEDGMENT

Special Thanks to:

California State University - Bakersfield

Department of Computer and Electrical Engineering and Computer Sciences

CSUB Fab Lab & Fab Lab Annex

Walter Stiern Library

&

Dr. Wei Li

THANK YOU

Does anyone have any questions?

REFERENCES

[1] Chu, Y.; Ho, C.; Lee, Y.; Li, B. Development of a Solar-Powered Unmanned Aerial Vehicle for Extended Flight Endurance. Drones 2021, 5, 44. https://doi.org/10.3390/drones5020044

[11] Agarwal, Akarsh. "DIY Radio Controller for Drone | nRF24L01." Hackster.io, 7 August 2021, DIY Radio Controller for Drone | nRF24L01 - Hackster.io

[12] Agarwal, Akarsh. "DIY Arduino Based Quadcopter Drone." Arduino Project Hub, 5 Sept. 2021, DIY Arduino based Quadcopter Drone - Arduino Project Hub.

[13] AkarshA2. "Flight Controller Tutorial | Arduino Based Quadcopter Drone." Instructables Circuits, Date Accessed: 2 Feb. 2022, Flight Controller Tutorial | Arduino Based Quadcopter Drone : 5 Steps - Instructables

[14] Hosseini, Saeed. "Arduino L293D Motor Driver Shield Tutorial." Electro Peak, Date Accessed: 2 Feb. 2022, Arduino L293D Motor Driver Shield Tutorial - ElectroPeak