COGNIZANCE ‘18

EVENT : Quadrone

Abstract

Institute of Radiophysics and Electronics (ECE), Calcutta University

Team Members :-

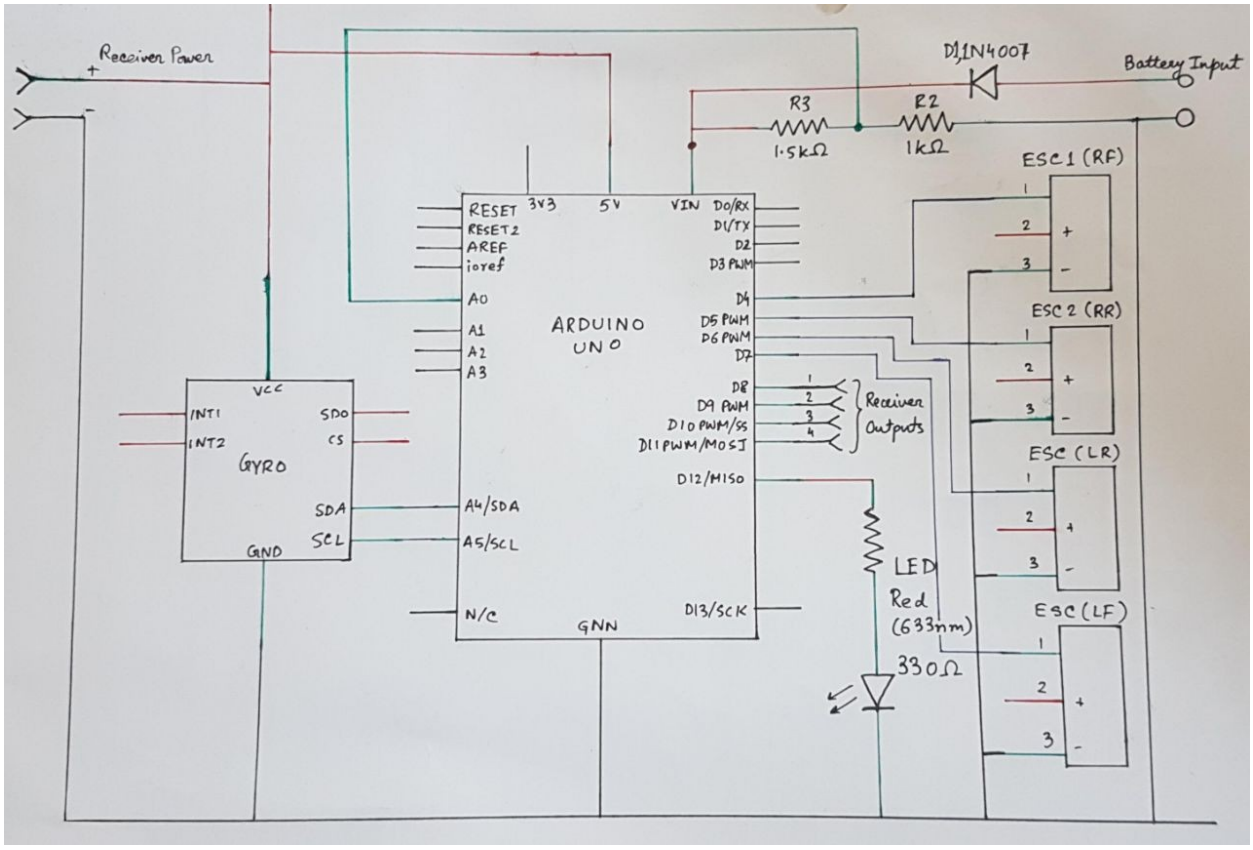
- **AANKIT DAS(**cogni1808754)
- **ABEER BANERJEE(**cogni1808767)
- **SINJOY SAHA (**cogni1809107)
- **SOUMYABROTO BANERJEE** (cogni1806812)

Materials required :-

- Quadcopter frame
- Brushless DC Motors (1000KV)
- ESCs (30A)
- Propellers (8"x4.5")
- Lipo battery (2200mAh, 11.1V)
- Arduino Uno
- MPU-6050*
- Radio Transmitter and Receiver (Flysky Model Number : FSCT6B)
- LED (red)
- Diode (1N4007)
- Resistors

***Note: The MPU-6050 3-axis gyro-accelerometer IC has been used as the 2-axis stabilisation unit was unavailable at the time and location of making the quadcopter. However, we haven't used the 3rd axis input from the MPU-6050. If need arises, we may show in Arduino code the 3rd axis input is disabled.**

SCHEMATIC :-


Receiver Power
+
-
D1 IN4007
Battery Input
R3
1.5kΩ
R2
1kΩ
ESC 1 (RF)
ESC 2 (RR)
ESC (LR)
ESC (LF)
RESET
3V3
5V
VIN
D0/RX
D1/TX
D2
D3 PWM
RESET 2
AREF
ioref
A0
A1
A2
A3
ARDUINO
UNO
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
N/C
Receiver outputs
VCC
INT1
INT2
SDO
CS
SDA
SCL
GND
GYRO
A4/SDA
A5/SCL
LED
Red
(633nm)
330Ω

Procedure : -

- Hardware

- All hardware components are assembled as per guidelines.
- The electronics are put together and soldered as the above schematic. For temporary connections wire connectors and jumper cables are used.
- The diode prevents current from the battery to flow into the USB while the Arduino is in Serial connection and cause a burnout.
- The LED serves as a low-power output of the battery indicator.
- The propellers are of 8" instead of the traditional 10" with 1000kV BLDC motors to shed some load off the motors and also to have quicker response during operations at higher speeds.
- The motors on the main diagonal rotate clockwise and vice versa on the other to prevent unnecessary yawing.

- Software

- All programming is done on the Arduino IDE.
- The setup() function sets up the Arduino as a flight controller by taking the necessary parameters like the Tx-Rx connection and the ESCs into account.
- The main loop() function does the following functions:
  1. Takes input from the gyro-accelerometer.
  2. We have sought to Read and Write directly via Registers referring to the Arduino Uno Data Sheet instead of the usual, conventional way of pinMode() function.
  3. Takes command from the radio receiver (which receives signals from the user controlled transmitter).
  4. Tries to equate the two inputs, for instance if the user wants a -5° pitch, the gyro should also be sending -5° pitch, letting us know that the quadcopter carried out the required operation.
  5. We have used a Complimentary Filter instead of Kalman Filter, for the complexity faced by using a Kalman Filter.
  6. Before the desired values are sent to the ESCs, they get weighted by a PID (proportional-integral-differential) controlling step, which is necessary for the stabilisation and fluid handling of the quadcopter.
  7. The PID gains are calibrated by a Trial and Error Mechanism till Stability is reached.
  8. The PID constants are then calculated using the PID equation and summed up under PID variables for a certain MPU controlled data.
  9. $\text{PID}=K_p * \text{error}(t) + K_i \int_0^t error(\tau)\,d\tau + K_d \frac{d(error(t))}{dt}$
  10. The weighted values are sent to ESCs, the response is checked .

- The source code(s) includes the following **HEADER** files:
    1. *Servo.h*
    2. *Wire.h*
    3. *Eeprom.h*

PROJECT FUNDED BY :-

- INSTITUTE OF RADIOPHYSICS AND ELECTRONICS, CALCUTTA UNIVERSITY
- TEQIP-PHASE III, CALCUTTA UNIVERSITY

VOTE OF THANKS :-

In the end we would like to thank all our Professors especially our Head of the Department, Dr.Debatosh Guha who we are lucky to have as a part of our daily life, all of those whose advices were unputdownable, last but not the least to seniors who constantly helped us through thick and thin, being the kernel of insipiration we had in our journey.

Picture of our Model :-