International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Design and Fabrication of Emergency Drone Recovery System

Aravindadhith. P. S

Abstract**:** ***The drone market has seen a surge in the past few years and the number of consumer drones has been on the increase. With even beginners starting to fly drones, there is a serious safety issue of whether the drone will crash because of technical and unforeseen problems. It may lead to injury of people going about their day and may even cause damage to material things on the ground and naturally damage the drone itself. So, there is a need for a new system which would substantially prevent these kinds of problems. The Emergency Drone System (EDR) will be a solution and will be a versatile fit in all drones. The EDR system uses the MPU-6050 gyroscope and accelerometer sensor along with MPL3115A2 Pressure based Altitude sensor to get input on the real-time movement of the drone and analyses using the algorithm coded into the Raspberry Pi using Python programming language. Apart from this, it also inputs data from the flight controller and analyses it. The parameters such as the altitude, rate of rotation and rate of fall are calculated and based on the algorithm if any kind of anomaly is detected it is programmed to deploy the parachute which will automatically prevent the crashing of the drone. So by which the crashing of drone and damage to property is prevented and safety of people below the drone is ensured.***

Keywords: Drone, recovery system, safety

I. INTRODUCTION

This system is designed and programmed in such a way that it could save drones that experience any failure in the avionics systems present on board or any motor failure. As the number of drones as increasing the need for safety is also arising, so many accidents are related to drone crashes. When drones are flying over people any failure in the avionics or motor failure will directly affect the people below them and it also will damages the drone. As the drones available in the market are usually more than $1000 there is a requirement to include safety features for it. Though they have abilities to land automatically when battery level is low, there is no feature or system available to protect the drone during any unexpected power failure or motor failure. So we made a system that can be attached on to the drone which can solve the above mentioned issue. The EDR system uses a complex pre-defined values to check whether the flying of drone is normal or not. If it finds any anomalies in the flying like sudden drop or sways the system is programmed to deploy the on board parachute by which the drone will be prevented from falling off. This reduces the impact that the drone absorbs when it hits the ground. This ensures the safety for both the drone and people below it and reduce the impact. This is an important feature or system that should be mandatory as accidents due to drones are increasing. This also prevents rogue behavior of drones as any anomalies in the flying and will deploy the parachute and stop the drone from flying away or crashing.

In the case of failure, the papers will be declined from the database of journal and publishing house. It is noted that: 1. Each author profile along with photo (min 100 word) has been included in the final paper. 2. Final paper is prepared as per journal the template. 3. Contents of the paper are fine and satisfactory. Author (s) can make rectification in the final paper but after the final submission to the journal, rectification is not possible.

Revised Manuscript Received on May 25, 2020.
* Correspondence Author
Aravindadhith*. P. S, Engineer, St. Joseph's Institute of Technology affiliated to Anna University, Chennai, Tamil Nadu, India.

II. LITERATURE REVIEW

Pooja Srivastava (etal.) [1]. This paper deals on operation of a rescue quadcopter using a Raspberry pi, the total drone is controlled using the Raspberry pi. This helps us with understanding of the basic operations using the Raspberry pi since our idea also deals with using Raspberry pi it gives an idea of interfacing the sensor with the Raspberry pi and implementing the idea. This paper also helps us in understanding the components in a quadcopter.

Anurag Singh Rajpoor (etal.) [2] This paper enlightens us with the quadcopter based on Arduino board, since Arduino is also similar to Raspberry pi but available at a cheaper rate so it provides idea of further implementing the idea onto the Arduino board and try to reduce the cost further more. This paper provides us with the connection diagrams of the circuits used also.

Basem AL-Madani (etal.) [3] In this paper the authors have implemented a parachute system using a fall detection algorithm based on MPU-6050 sensor so that during any failures the system will deploy the parachute within 0.5 seconds. The system also used Kalman filter to calibrate the sensor to compensate the loses for temperature changes. This paper greatly helps us with our idea, but the drawback in this paper is that when the sensor fails to detect the fall or undesired detection occurs there is no system to check the reliability of the signal. So if the sensor detects an anomaly even when the drone is flying properly it deploys the parachute causing undesired results.

III. PROBLEM DEFINITION

The drones are priced at a high rate in the present market scenario and the drone crashes are also increasing.

The crashes result in damage to property or the drone, at times even humans are injured. This may be due to the easy access to drone to the hands of common people who have not taken formal training for drone flying, in case of careless flying it leads to drone crashes. So, there is a need for improving the safety of drones and people.

IV. METHODOLOGY

The Drone Recovery System basically uses signals that happen during a failure of a drone. The Raspberry pi operates on a separate circuit from the main flying circuits and obtains data from the sensors and the flight controller rather than operating on the same circuit. The data obtained from the sensors are analyzed against the reference value and the based on the algorithm the decision is taken. This is to provide a failsafe mechanism even when the main circuitry of the drone fails, the EDRS is capable of deploying the parachute at these kind of situations.

A. Sensors Used

i. MPU-6050

The device uses a MPU-6050 Gyro-Accelerator sensor. The MPU-6050 incorporates InvenSense's MotionFusion™ and run-time calibration firmware that enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices in motion-enabled products, guaranteeing that sensor fusion algorithms and calibration procedures deliver optimal performance. The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor™ (DMP™), which processes complex 6-axis MotionFusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I²C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor.

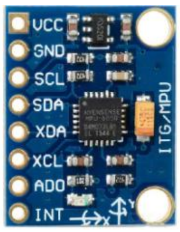

Fig. 1. MPU-6050 Sensor [4]

ii. MPL3115A2

The MPL3115A2 is a compact, piezoresistive, absolute pressure sensor with an I2C digital interface. MPL3115A2 has a wide operating range of 20 kPa to 110 kPa, a range that covers all surface elevations on earth. The MEMS is temperature compensated utilizing an on-chip temperature sensor. The pressure and temperature data is fed into a high resolution ADC to provide fully compensated and digitized outputs for pressure in Pascals and temperature in °C. The compensated pressure output can then be converted to altitude, utilizing the formula:

$$h = 44330.77\left\{1-\left(\frac{p}{po}\right)^{0.1902632}\right\} + OFF\_H(Register\ Value) \qquad (1)$$

where:
*p*0 = sea level pressure (101,326 Pa)
h = altitude in meters
"Pressure/altitude" provided in meters.The internal processing in MPL3115A2 removes compensation and unit conversion load from the system MCU, simplifying system design. The MPL3115A2 sensor provides the Pi with the altitude of the drone, so by which any failure in the drone will make the drone to fall by which the altitude will drop and it can be sensed by this sensor.

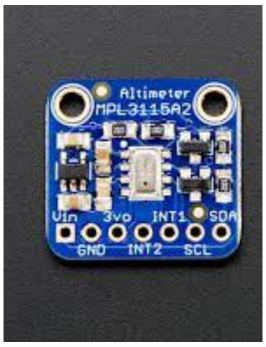

Fig. 2. MPL3115A2 Sensor [5]

B. Circuits Used

i. Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

V. OPERATION

The sensors are connected to the Raspberry Pi using jumper cables they Raspberry Pi is powered using a separate 5-Volt battery, and not to the main LiPo battery since in case of failure of LiPo battery the EDRS should continue to function. The MPU-6050 sensor and a jumper cable taken parallel from the flight controller are the two inputs that are to be connected to the Pi board. From which the serial GPIO pins in the Pi are capable of reading the signals from it. The MPL3115A2 sensor provides the Pi with the altitude of the drone, so by which any failure in the drone will make the drone to fall by which the altitude will drop and it can be sensed by this sensor. Also, the input which we obtain from the flight controller is used to check whether the system is has power if not the command to deploy the parachute is executed. Since most failures occur due to the power supply failure to the flight controller. Before deploying the parachute the Pi board sends a signal to the RC controller that the parachute is about to be deployed. So it provides a 0.5 second delay before deploying the parachute in order to cancel the deployment. This provided to prevent any accidental deployment of the parachute. The input to cancel the deployment is obtained from the RC using one of the available channels.

If no input is received the parachute is deployed. This feature can be used for troubleshooting also.

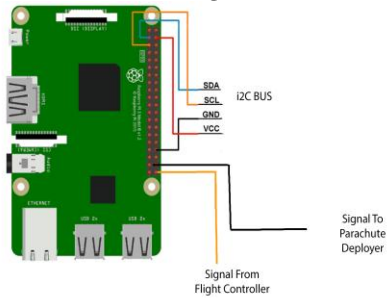

Fig. 3. Circuit Diagram [6]

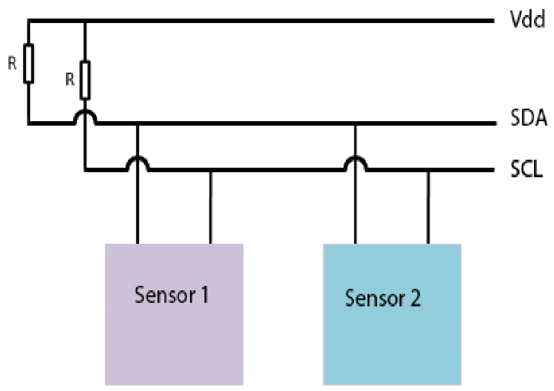

Fig. 3. I2C BUS Circuit Connection

VI. RESULTS AND DISCUSSION

i. MPU-6050 Sensor Readings

The data readings from the MPU-6050 sensor at the hovering position and in movement were obtained, these data will be serving as reference value for deploying the parachute since the parachute can only be deployed if the reading become abnormal. The raw values obtained from the MPU-6050 sensor at hovering position is :

Table- I: Reading from MPU-6050 (Hovering)

Parameters	Sensor Output
X-Axis Acceleration	20
Y-Axis Acceleration	30
Z-Axis Acceleration	74
X-Axis Gyroscope	1642
Y-Axis Gyroscope	128
Z-Axis Gyroscope	14944

The data obtained from MPU-6050 sensor when drone is in motion is as follows.

Table-II : Reading from MPU-6050 (In Motion)

Direction of motion	X-Axis Acceleration	Y-Axis Acceleration	Z-Axis Acceleration	X-Axis Gyro	Y-Axis Gyro	Z-Axis Gyro
Forward	-22	29	73	1396	-4064	14308
Reverse	-27	27	73	1040	4528	14324
Left	-24	29	70	6452	96	13756
Right	-23	27	71	4636	148	13648

ii. Rate of Descent calculation for free fall

The rate of descent by a drone when controlled by the motors will be lesser than the rate of descent during free fall, this is because the motors provide with a minimum amount of thrust even when the throttle is at the bottom position. So, we need to calculate the rate of decent during the free fall condition. We could use the equation of motion formula in order to find out the values. This would provide us with a free fall time in vacuum, but in actual scenario the air resistance will increase the free fall time, we are considering the worst case scenario for our idea. The formula to calculate is as follows:

$$s = (u * t) + (\frac{1}{2} * a * t^2) \tag{2}$$

$$t = \sqrt{\left(2 * \frac{(s-(u*t))}{a}\right)} \tag{3}$$

From the formula (3) we could find out the rate of descent for certain drop heights. In the formula the initial velocity (u) is zero since the fall is considered to be free fall. The acceleration (a) value is taken to be 9.81 m/s, the acceleration due to gravity.

Table-III : Time for Free-Fall

Height (meters)	Time taken for drop (seconds)	Time assumed for algorithm (seconds) (10% safety factor)
3	0.7821	0.86031
5	1.0096	1.11056
7	1.1946	1.31406
10	1.4278	1.57025

Table-IV : Time taken for landing using motors

Height (meters)	Time taken (seconds)
5	1.9587
10	2.7785

So, from the table we could obtain certain time values for the heights, so these values can be used for the algorithm. So if the drone descends the height in less than time range mentioned in the column 2 of table 3 and within the free fall time then the parachute deployment signal is relayed.

iii. Algorithm

The EDRS follows an algorithm in order to decide when to deploy the parachute. This is important as a single command must not be relied upon for deployment rather a feedback looped system needs to be used to check with all the sensors available and deploy the parachute.

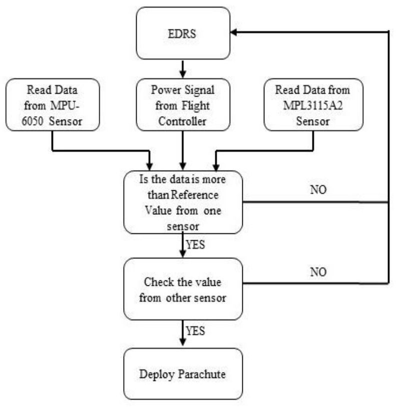
EDRS
Read Data from MPU-6050 Sensor
Power Signal from Flight Controller
Read Data from MPL3115A2 Sensor
Is the data is more than Reference Value from one sensor
NO
YES
Check the value from other sensor
NO
YES
Deploy Parachute

Fig. 4. Algorithm Flow Chart

The above flow chart shows the algorithm based on how the decision to whether deploy or not the parachute is taken. The algorithm does have redundancy this is to overcome the previous problem where a single sensor may not be reliable to be used. But in this system when receiving a signal from a sensor the data from the other sensor also is cheeked whether any anomaly is present if both the sensors detect the falling of drone the command to deploy the parachute is sent and deployment takes place.

iv. Parachute Selection [1]

The parachute was made from a rugged nylon material. Parachute dimensions are chosen such that a 2 kg drone does not fall faster than 5 m/s. Under the selected dimensions, a 16-segment dome-shaped parachute with overlock stitches was chosen. The diameter of the parachute is 1.45 m.

VII. CONCLUSION

The EDRS provides solution for the drone crashing problem and helps in preventing the drone from crashing by using both the MPU-6050 and MPL3115A2 sensor to deploy the parachute when an anomaly is detected. The usage of 2 sensors helps us with redundancy and only deploys the parachute if the drone is about to crash, preventing false positives. By this way it overcome the problem of usage of single sensor, in which if the sensor fails at any point it will lead the drone to crash. Considering the safety factors included for the calculation of the free fall time, the EDRS will definitely will help in preventing the crashing of drones.

REFERENCES

1. Basem AL-Madani, Marius Svirskis, Gintautas Narvydas, RytisMaskeli0nas , and Robertas Damaševiľius., Design of Fully Automatic Drone Parachute System with Temperature Compensation Mechanism for Civilian and Military Applications., Journal of Advanced Transportation., Volume 2018, Article ID 2964583, 11 pages.,
2. Anurag Singh Rajpoot, Namrata Gadani, Sagar Kalathia., Development of Arduino Based Quadcopter., International Advanced Research Journal in Science, Engineering and Technology Vol. 3, Issue 6, June 2016., ISSN (Online) 2393-802., ISSN (Print) 2394-1588. PP 252-259
3. Pooja Srivastava, Tejaswi Ninawe, Chitral Puthran, Vaishali Nirgude., Quadcopter for Rescue Missions and Surveillance IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727 PP 48-52.
4. MPU-6050 sensor image source www.elementzonline.com/image/cache/catalog/data/products/Sensors%20and%20Modules/Accelerometers/MPU6050/sku_154602_2-500x500.jpg
5. MPL3115A2 sensor image source https://cdn-shop.adafruit.com/1200x900/1893-00.jpg

AUTHORS PROFILE

Aravindadhith. P. S is an undergraduate engineer who completed his degree from St.Joseph's Institute of Technology affiliated to Anna University. He is an avid drone enthusiast and drone pilot.