Fayoum University

Engineering Faculty

Electrical Engineering Department

B. Eng. Final Year Project

Quadcopter (drive test)

By:

Bassem Adel Adly

Luka Ayad Michael

Ramez Maged Adly

Mina Makram Elia

Supervised By:

Amr M. Gody , professor

Supervisor(s) *Date of examination*

١٠/٧/٢٠١٦

DEDICATION

We dedicate our work to our families.

ACKNOWLEDGMENT

We want to acknowledge the contribution of Dr/ Amr M. Gody for supervising and helping us in the project and the facilitations that he offered and also the spirit of joy and fun during the work.

We also want to acknowledge the contribution of our teacher assistants, Eng/Ahmad Hamdy and Eng/Maichal Refaat for leading us through the beginning of our project.

We want to thank Dr/ Amr M. Gody , Dr/ Rania Foaad and Mr/Abd Al Razek for allowing us using the laboratories of our faculty.

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Bachelor of Science in *Electrical Engineering* is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

Registration No.: _____

Date: _____

ABSTRACT

Making the mobile telecommunication drive test an easier process. When they build a mobile base station, they want to practically test the mobile coverage that takes many days and effort through the traditional drive test process. By making the drivetest through the quad copter, the process will much easier, effective, time efficient and also fun.

Table of content

Contents

LIST of FIGURES

Chapter One

١. INTRODUCTION

Drones belong to a class of aerial vehicles known as Unmanned Aerial Vehicles (UAVs). These vehicles can take to the air without pilots.

The four-rotor helicopter designed by Louis Breguet. This was the first rotary wing aircraft to lift itself off the ground, although only in tethered flight at an altitude of a few feet. In ١٩٠٨, it was reported as having flown 'several times', although details are sparse.

In the last few decades, small-scale unmanned aerial vehicles have been used for many applications. The need for aircraft with greater maneuverability and hovering ability has led to a rise in quad copter research. The four-rotor design allows quad copters to be relatively simple in design yet highly reliable and maneuverable.

Research is continuing to increase the abilities of quad copters by making advances in multi-craft communication, environment exploration, and maneuverability. If these developing qualities can be combined, quad copters would be capable of advanced autonomous missions that are currently not possible with other vehicles.

Essentially, this makes drones flying robots. Encompassing both planes and quadcopters, they have software-controlled flight plans integrated into their systems. These systems work with Global Positioning Technology (GPS) to guide and track their movements.

Chapter Two

٢. SYSTEM LAYOUT

٢.١ Project layout

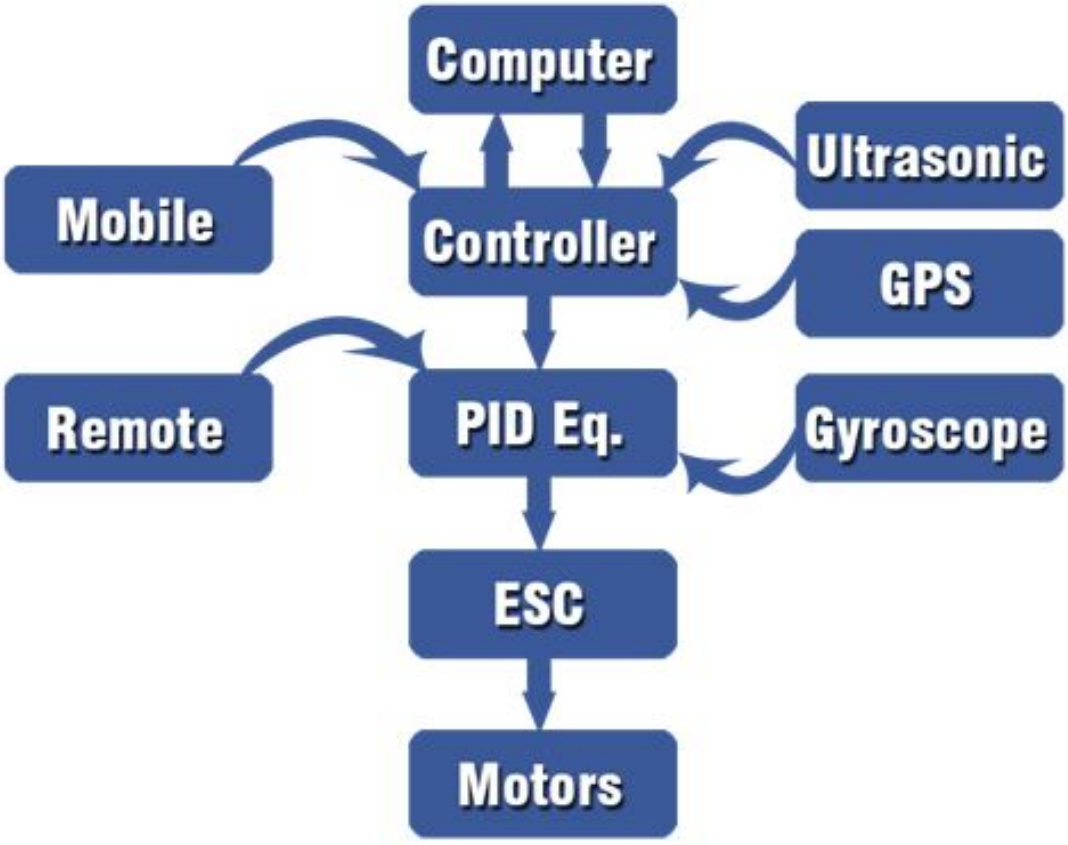


Figure ١ System Layout

٢.٢ Main PID Program flow

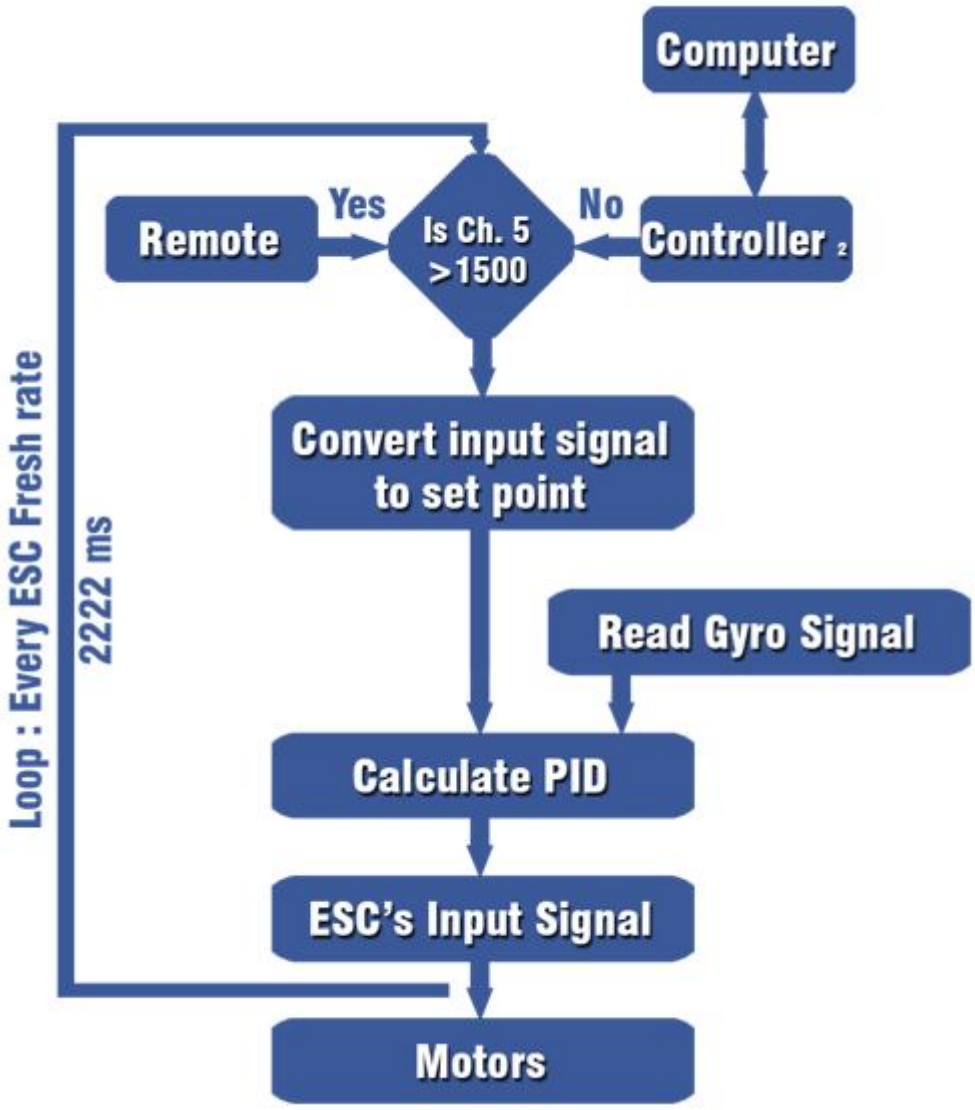


Figure ١ PID Program Flow

Chapter Three

٣. HARDWARE COMPONENTS

In this project, there are many hardware components used. There are microcontrollers used for controlling the project like Atmega١٣٨ (in Arduino Nano) and Atmega٢٥٦٠ (in Arduino Mega). There are the Quad Copter main parts like Brushless motors, Aluminum frame, Propellers, Electronic speed controllers (ESC), LiPo Battery and Power distribution board. There are many made Voltage dividers to measure the battery in copter. There are also many sensors used like Gyroscope, GPS and Ultrasonic. There are two ways of communicating RC Remote and Receiver, also there are two APC٢٢٠ Transceivers. In addition to the end user devices for applications like Computer and Android phone. In this chapter, these components are discussed in more details.

٣.١ Atmega٣٢٨ (Arduino Nano)

The Arduino Nano is a very small, complete, and breadboard-friendly board based on the ATmega٣٢٨ (Arduino Nano ٣.٠). Arduino Nano board is very suitable for applications where small and compact size is important for user.

Figure ٢ Arduino Nano

Specifications

- Microcontroller ATmega٣٢٨
- Operating Voltage ٥ V
- Input Voltage ٧-١٢ V
- Digital I/O Pins ١٤ (of which ٦ provide PWM output)
- Analog Input Pins ٨
- Flash Memory ٣٢ KB (ATmega٣٢٨)
- SRAM ٢ KB (ATmega٣٢٨)
- EEPROM ١ KB (ATmega٣٢٨)
- Clock Speed ١٦ MHz
- Dimensions ١٨.٥ mm x ٤٣.١٨ mm

ATmega ٣٢٨ feature

- Peripheral Features:
  – Two ٨-bit Timer/Counters with Separate Prescaler and Compare Mode
  – One ١٦-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  – Real Time Counter with Separate Oscillator
  – Six PWM Channels
  – ٨-channel ١٠-bit ADC in TQFP and QFN/MLF package
  Temperature Measurement
  – ٦-channel ١٠-bit ADC in PDIP Package
  Temperature Measurement
  – Programmable Serial USART
  – Master/Slave SPI Serial Interface
  – Byte-oriented ٢-wire Serial Interface (Philips I٢C compatible)
  – Programmable Watchdog Timer with Separate On-chip Oscillator
  – On-chip Analog Comparator
  – Interrupt and Wake-up on Pin Change

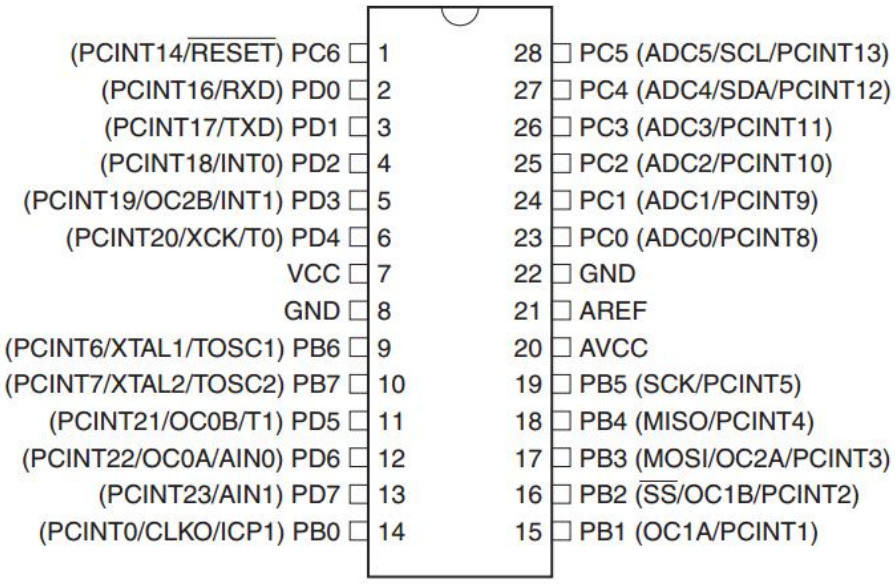


Figure ٣ Atmega٣٢٨ pin diagram

٣.٢ Atmega٢٥٦٠ (Arduino Mega)

The Arduino Mega is a microcontroller board based on the Atmega٢٥٦٠. It has ٥٤ digital input/output pins (of which ١٤ can be used as PWM outputs), ١٦ analog inputs and ٤ UARTs (hardware serial ports) with a ١٦ MHz crystal oscillator.

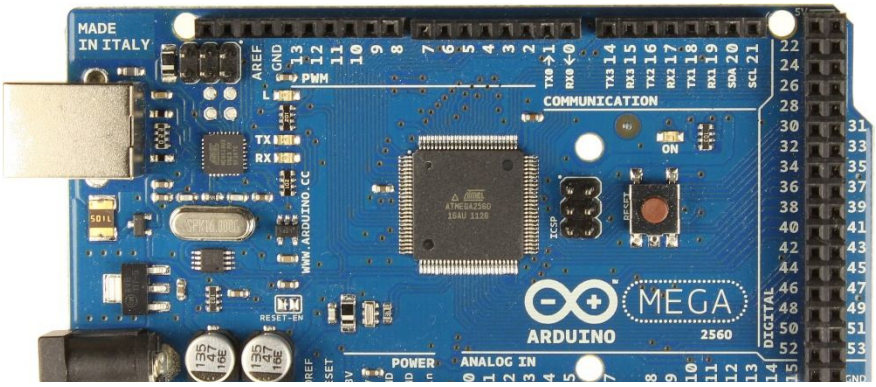


Figure ٤ Arduino Mega R٣

Technical specs

Microcontroller	ATmega٢٥٦٠
Operating Voltage	٥V
Input Voltage (recommended)	٧-١٢V
Input Voltage (limit)	٦-٢٠V
Digital I/O Pins	٥٤ (of which ١٥ provide PWM output)
Analog Input Pins	١٦
DC Current per I/O Pin	٢٠ mA
DC Current for ٣.٣V Pin	٥٠ mA
Flash Memory	٢٥٦ KB of which ٨ KB used by bootloader
SRAM	٨ KB
EEPROM	٤ KB
Clock Speed	١٦ MHz
Length	١٠١.٥٢ mm
Width	٥٣.٣ mm
Weight	٣٧ g

ATmega٢٥٦٠ Features

Peripheral Features

–Two ٨-bit Timer/Counters with Separate Prescaler and Compare Mode

–Four ١٦-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode

–Real Time Counter with Separate Oscillator

–Four ٨-bit PWM Channels

–Twelve PWM Channels with Programmable Resolution ١٦ Bits

–Output Compare Modulator

-١٦channel, ١٠-bit ADC

–Four Programmable Serial USART

–Master/Slave SPI Serial Interface

–Byte Oriented ٢-wire Serial Interface

–Programmable Watchdog Timer with Separate On-chip Oscillator

–On-chip Analog Comparator

– Interrupt and Wake-up on Pin Change

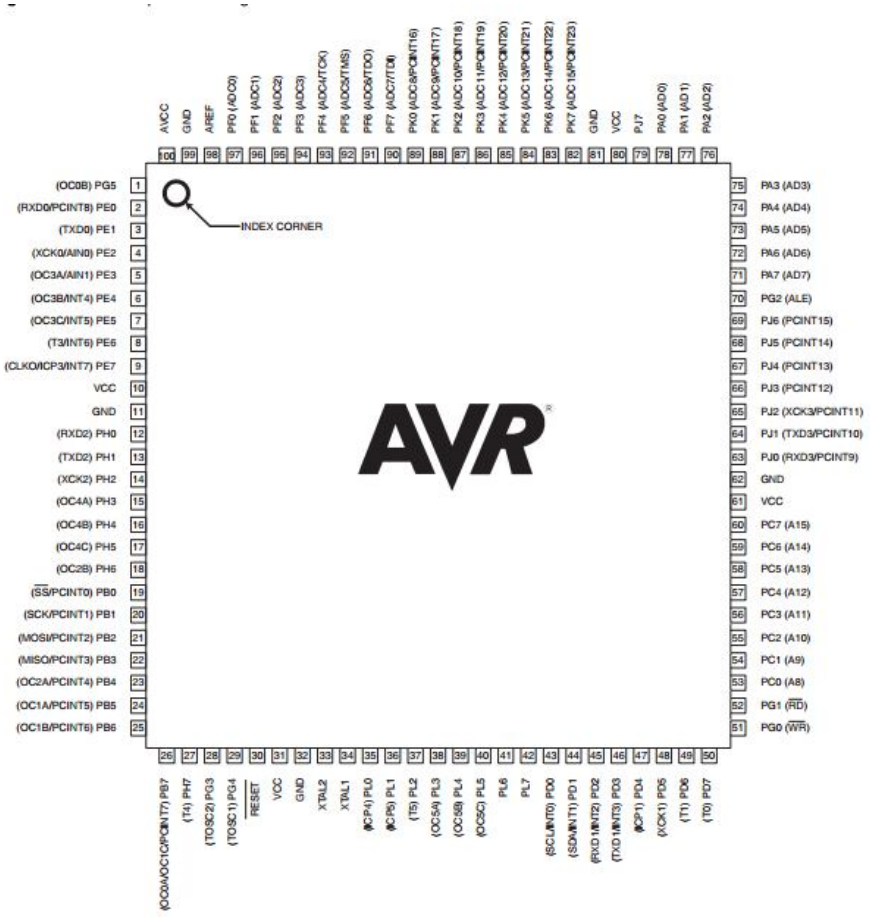


Figure ٥ Atmega ٢٥٦٠ pin diagram

٣.٣ Brushless motors

Brushless DC electric motor (BLDC motors, BL motors) also known as electronically commutated motors (ECMs, EC motors) are synchronous motors that are powered by a DC electric source via an integrated inverter/switching power supply, which produces an AC electric signal to drive the motor. In this context, AC, alternating current, does not imply a sinusoidal waveform, but rather a bi-directional current with no restriction on waveform. Additional sensors and electronics control the inverter output amplitude and waveform (and therefore percent of DC bus usage/efficiency) and frequency (i.e. rotor speed).

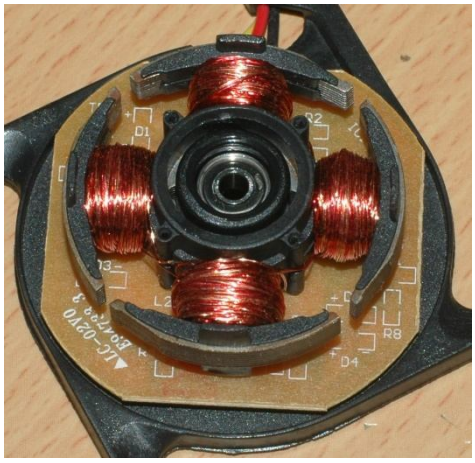


Figure ٦ internal coils of brushless motors

The rotor part of a brushless motor is often a permanent magnet synchronous motor, but can also be a switched reluctance motor, or induction motor.

Brushless motors may be described as stepper motors; however, the term "stepper motor" tends to be used for motors that are designed specifically to be operated in a mode where they are frequently stopped with the rotor in a defined angular position.

Thrust force is a function of:

- Pitch
- Diameter
- Material
- RPM: Revolutions per minute (abbreviated rpm, RPM, rev/min, r/min), is a measure of the frequency of rotation, specifically the number of rotations around a fixed axis in one minute. It is used as a measure of rotational speed of a mechanical component.
- RPM = KV * Total Battery Voltage
- Where:
- KV: Motor velocity constant of an electric motor.
- Temp

Motor which is used:

Quanum MT series motors are Spec built by DYS, well known in the industry for their
quality and performance.

The MT line uses quality NMB bearings, high pole counts for torque and smooth operation and all have standardized universal mountings in their given size.

Quanum has spec'd the motors with N٤٢sh magnets and .٢mm laminations for optimal
efficiency. The MT's also have ample motor wire length for a cut to fit installation, with included bullet connectors and shrink tube. Another great feature is that Quanum spec'd all the motors to come with mounting hardware in various lengths, as well as both hub and direct mount propeller options, so installation is a snap.

Each Quanum motor is QC checked for balance, wind, performance and mechanical

Figure ٧ Quanum Brushless motor MT٢٢١٧

tolerance.

Features:

• High Power N٤٢sh magnets

• ٠.٢mm laminations for optimal efficiency

• Ample cut to length motor wires for a clean install

• Direct mount propeller or Hub style compatible

• Multilength mounting hardware included

Specs:

- KV(RPM/V): ٨٠٠KV
- Lipo cells: ٣ - ٥S
- Max Power: ٢٨٤.٩W
- Max Amps: ١٥.٤A
- No Load Current: ٠.٧٤/١٨.٥V
- Internal Resistance: ٠.٤٤ohm
- Number of Poles: ١٤
- Dimensions(Dia.* L): ٢٧.٥*٣٢.٥mm
- Motor Shaft: ٣mm
- prop shaft: ٦mm CW prop adapter
- Weight: ٦٨g
- bolt hole spacing: ١٦*١٩ M٣
- Lamination thickness: ٠.٢mm
- Magnets: ٤٢SH
- Wire: ١٨AWG
- Connector: ٣.٥mm bullet

٣.٤ Aluminum frame

Frame is the structure that holds all the components together. The Frame should be rigid, and be able to minimize the vibrations coming from the motors.

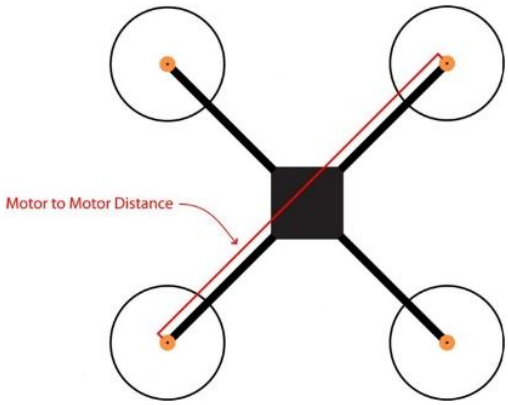


Figure ٨ Model of quadcopter frame

A QuadCopter frame consists of two to three parts, which don't necessarily have to be of the same material:

١- The center plate where the electronics are mounted
٢- Four arms mounted to the center plate
٣- Four motor brackets connecting the motors to the end of the arms

Most available materials for the frame are:

١- Carbon Fiber
٢- Aluminum
٣- Wood, such as Plywood or MDF (Medium density fiberboard)

Frame which is used:

Hollow aluminum square rails is the most popular for the QuadCopters' arms due to its relatively light weight, rigidness and affordability. However aluminum could suffer from motor vibrations, as the damping effect is not as good as carbon fiber. In cases of severe vibration problem, it could mess up sensor readings.

As for arm length, the term "motor to motor distance" is sometimes used, meaning the distance between the center of one motor to that of another motor of the same arm in the

QuadCopter terminology. The motor to motor distance usually depends on the diameter of the propellers. To make you have enough space between the propellers and they don't get caught by each other.

٣.٥ Propellers

On each of the brushless motors, there is a mounted propeller.

Thefour propellers are actually not identical. The front and the back propellers are tilted to the right, while the left and right propellers are tilted to the left. Two rotors rotates in the opposite directions to the other two to avoid body spinning. By making the propeller pairs spin in each direction, but also having opposite tilting, all of them will provide lifting thrust without spinning in the same direction. This makes it possible for the Quad Copter to

Figure ٩ Propellers

stabilize the yaw rotation, which is the rotation around itself.

The propellers come in different diameters and pitches (tilting). The decision to use one is according to the frame size.

- Some of the standard propeller sizes used for Quad Copters are:

- EPP١٠٤٥ ١٠ diameter and ٤.٥ pitch this is the most popular one, good for midsized quads
- APC ١٠٤٧ ١٠ diameter and ٤.٧ pitch much similar to the one above
- EPP٠٨٤٥ ٨ diameter and ٤.٥ pitch regularly used in smaller quads
- EPP١٢٤٥ ١٢ diameter and ٤.٥ pitch used for larger quads which requires lot of thrust
- EPP٠٩٣٨ ٩ diameter and ٣.٨ pitch used in smaller quads

General rules in selecting propellers:

- The larger diameter and pitch the more thrust the propeller can generate. It also requires more power to drive it,
- When using high RPM (Revolutions per minute) motors, one should select the smaller or midsized propellers. When using low RPM motors, one should select the larger propellers as troubles can accur with the small ones not being able to lift the quad at low speed.

Analysis of Propeller Pitch, Diameter, and RPM

Pitch VS Diameter: the diameter means area while pitch means effective area. Therefore, with the same diameter, larger pitch propeller would generate more thrust and lift more weight but also use more power.

A higher RPM of the propeller will give more speed and maneuverability, but it is limited for weight it will be able to lift for any given power. Also, the power drawn (and rotating power required) by the motor increases as the effective area of the propeller increases, so a bigger diameter or higher pitch one will draw more power at the same RPM, but will also produce much more thrust, and it will be able to lift more weight.

٣.٦ Electronic speed controllers (ESC)

The brushless motors are multi-phased, normally ٣ phases, so direct supply of DC power will not turn the motors on. That is where the Electronic Speed Controllers (ESC)

Figure ١٠ Electronic speed contoller (ESC))

comes into play.

The ESC generating three high frequency signals with different but controllable phases continually to keep the motor turning. The ESC is also able to source a lot of current as the motors can draw a lot of power.

The ESC is an inexpensive motor controller board that has a battery input and a three phase Output for the motor. Each ESC is controlled independently by a PPM signal (similar to PWM). The Frequency of the signals also vary a lot, but for a Quad copter it is recommended the controller Should support high enough frequency signal, so the motor speeds can be adjusted quick enough for optimal stability (i.e. at least ٢٠٠ Hz or even better ٣٠٠ Hz PPM signal). When selecting a suitable ESC, the most important factor is the source current. You should alwaysChoose an ESC with at least ١٠A or more in sourcing current as what your motor will require. Second most important factor is the programming facilities.

Features

- Current: ٣٠A dc current
- Work with: lipo battery ٢S-٤S
- Size(mm): ٥٣*٢٥*١١
- Weight(g): ٢١.١
- Weight(g): with line: ٣٠.٨

- Frequency: ٤٥٠ HZ
- the connection in the ESC

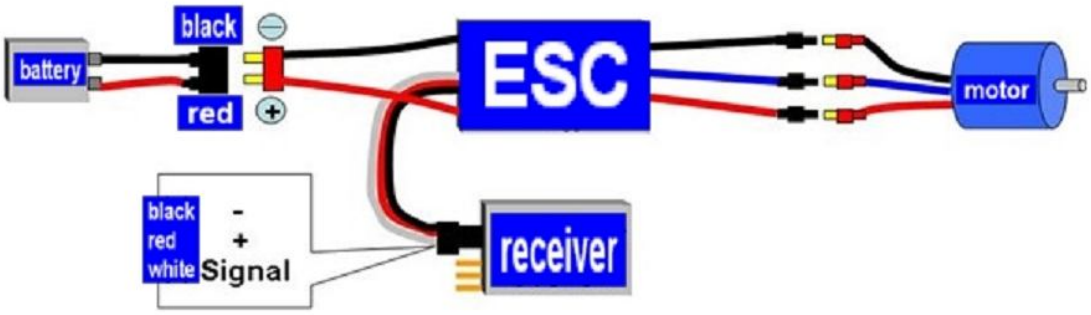


Figure ١٢ Simple connection between ESC, Battery and RC receiver

٣.٧ Power distribution board

This is a power distribution board (PDB) for any application that use one battery with with man outputs (such as quad-copter, which use one battery and ٤ brushless motors). It can be used to connect battery up to eight motors (hexa or octa-copter).

This power distribution board makes wiring a whole lot easier. The board mounts inside the frame using small spacers and screws. Simply solder the ESC directly to the board, and a battery lead to into two plated holes. The board lines up with existing center plate holes.

Figure ١٢ power distribution board

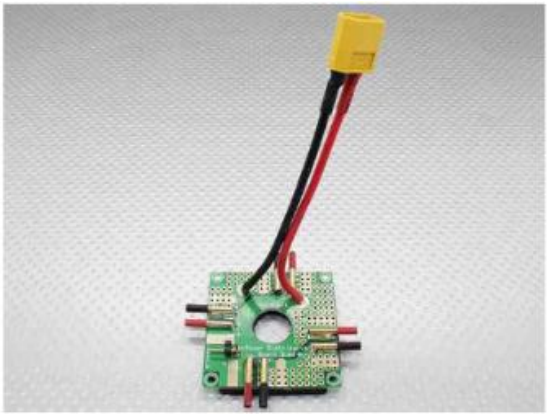


Figure ١١ power distribution board

٣.٨Battery

Figure ١٣ LiPo battery

Lithium Polymer Battery (١١.١ V, ٥٢٠٠ mAh- ٣٥C)

As for the power source of the quadcopter, I would recommend LiPo Battery because firstly it is Light, and secondly its current ratings meet our requirement. NiMH is also possible. They are cheaper, but it's also a lot heavier than LiPo Battery.

Features

- Capacity: ٥٢٠٠mAh
- Configuration: ٣S١P / ١١.١v / ٣Cell
- Peak Discharge: ٣٥C
- Pack Weight: ٤١٣g
- Pack Size: ٣٠.٦x٤٤x١٣٨
- Connector Type: T connector

Battery Discharge Rate

An important factor is the discharge rate, which is specified by the C-value. The C-value together with the battery capacity indicates how much current can be drawn from the battery.

٣.٩ Gyroscope

Figure ١٤ Gyroscope MPU ٦٠٥٠

IMU (٣ Axis Gyroscope + ٣ Axis Accelerometer) MPU-٦٠٥٠

The sensor contains a ٣-axis MEMS accelerometer and a three axis MEMS Gyro in a single chip. It is very accurate, since it contains ١٦-bits analog to digital conversion hardware for each channel. Therefore, it capturesthe x, y, and z channel at the same time. In addition, it is accurate because you have the ٣-axis gyro and ٣-axis

Features

- I٢C Digital-output of ٦ or ٩-axis motion Fusion data in rotation matrix, quaternion, Euler Angle, or raw data format
- Input Voltage: ٢.٣ - ٣.٤V.
- Tri-Axis angular rate sensor (gyro) with a sensitivity up to ١٣١ LSBs/dps and a full-scale range of ±٢٥٠, ±٥٠٠, ±١٠٠٠, and ±٢٠٠٠dps.
- Tri-Axis accelerometer with a programmable full scale range of ±٢g, ±٤g, ±٨g and ±١٦g
- Digital Motion Processing™ (DMP™) engine offloads complex Motion Fusion, sensor timing synchronization and gesture detection.
- Digital-output temperature sensor.

٣.١٠ GPS

Ublox NEO-٦m GPS Module

Figure ١٥ GPS Ublox neo ٦m

Ublox makes good gps. The Ublox NEO-٦M gps engine on this board is a quite good one, with high precision binary output. It has also high sensitivity for indoor applications. The gps module have a battery for power backup and EEprom for storing configuration settings.

The antenna is connected to module through ufl cable, which allow for flexibility in mounting the gps such that the antenna will always see the sky for best performance. This make it powerful to use with cars and other mobile applications.

The Ublox gps module has serial TTL output and it has four pins: TX, RX, VCC and GND.

Features

- Cold start time of ٣٨ s and Hot start time of ١ s
- Supply voltage: ٣.٣ V t
- Configurable from ٤٨٠٠ Baud to ١١٥٢٠٠ Baud rates. (default ٩٦٠٠)
- SuperSense ® Indoor GPS: -١٦٢ dBm. tracking sensitivity
- ٥Hz position update rate
- Operating temperature range: -٤٠ TO ٨٥°C
- UART TTL socket
- EEprom to save configuration settings

- Rechargeable battery for Backup
- Separated ١٨X١٨mm GPS antenna
- Dimension: ٢٢X٣٠X١٣ mm
- Weight: ١٢g

٣.١١ Ultrasonic

It is an IC that works by sending an ultrasound pulse at around ٤٠ KHz. It then waits and listens for the pulse to echo back, calculating the time taken in microseconds. You can trigger a pulse as fast as ٢٠ times a second and it can determine objects up to ٥ meters away and as near as ٣cm. It needs a ٥V power supply to run.

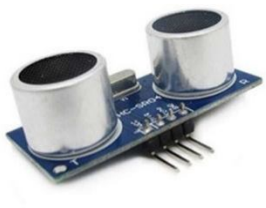


Figure ١٦ Ultrasonic range detection sensor

Specifications:

- Table ١: Specifications
- Working voltage ٥V(DC)
- Working current max ١٥ mA
- Working frequency ٤٠ KHZ
- Output signal ٠-٥V (high when obstacle in range)
- Sentry angle max ١٥ degree
- Sentry Distance ٢cm - ٥٠٠cm
- High-accuracy ٠.٣cm
- Input trigger signal ١٠µs TTL impulse
- Echo signal : output TTL PWL signal

٣.١٢ RC Remote and Receiver

Firstly, we use (flysky fs-t٦) transmitter and receiver - ٦ channel

Figure ١٧ flysky fs-t٦ remote transmitter

The transmitter consist of

- (Up and down) left stick is used to control throttle of the copter => channel ٣
- (right and left) left stick is used to control yaw angle of the copter => channel ٤
- (Up and down) right stick is used to control pitch angle of the copter => channel ٢
- (right and left) right stick is used to control roll angle of the copter => channel ١
- Switch D is used to change the control between computer and RC remote
- Switch C is used for emergency landing
- Power switch to on off the transmitter

The receiver consist of

- antenna
- ٦ output signal

The output signal from receiver:

PWM and PPM are two common words used in the R/C industry. PWM stands for Pulse Width Modulation and PPM stands for Pulse Position Modulation. Some devices that use PWM for control are ESC's (electronic speed controls) and servos. PWM is a technique used to relay data in the form of a varying pulse width.

You may be already familiar with binary, ١'s and ٠'s; where a ١ is represented as 'on' and a ٠ as 'off'. An example of this would be a light switch. Turning the switch on would indicate a ١, off a ٠. In the case of a PWM/PPM signal, a voltage applied indicates a ١ and vice versa. However, in the case of R/C electronics, this 'on/off' data is not enough; this is where the pulse width comes in.

The way we relay data to a servo for instance is the time the pulse is on. In the case of R/C electronics this time is usually around ١-٢ milliseconds. A servo or ESC will monitor this pulse and begin counting when the pulse is detected and stop counting when the pulse stops. The time the pulse is on will determine the servo position. For example, sending a servo a ١ms pulse will make the servo swing completely left while a ٢ms pulse will swing the arm completely right.

Generally, in R/C equipment an entire PWM pulse will last a total of ٢٠ms. The entire pulse is called a frame. A complete frame will include both the time the pulse is high (١-٢ms) and the time the pulse is low. The image below represents a typical PWM frame.

Although the frame lasts ٢٠ms the important part of the pulse is the time the pulse is on; ١-٢ms. Although the time between pulses is not as important it does play an important role. Usually keeping the time between pulses around ٢٠ms is best. If the delay is longer, a servo for example will lose holding power. A pulse can be generated much faster but ٢٠ms is best for most situations.

R/C Devices that use PWM Pulses:

- Servos
- Electronic Speed Controllers
- R/C switches
- R/C lights
- R/C receivers
- Data loggers
- Failsafe's
- Autopilot/Stabilization systems
- Servo Controller

٣.١٣ APC٢٢٠ Transceivers

APC٢٢٠ Wireless Kit

Figure ١٨ APC٢٢٠ wireless kit

This is APC٢٢٠ wireless kit; it includes two APC٢٢٠ modules, two external antennas and one usb-ttl converter. The APC٢٢٠ radio module provides a simple and economic Solution to wireless data communications. The employment of an embedded high-speed microprocessor and high performance IC creates a transparent UART/TTL interface, and eliminates any need for packetizing and data encoding.

APC٢٢٠-٤٣ is a cost-effective and easy applied module that not only can transmit transparent data with large data buffer zone. The parameters easily setting and small size make the module an ideal for wireless data transfer application.

The module is configured through an interface program called RF-Magic and a USB-TTL converter. The RF module fits in the USB-TTL converter, which is supplied with the set.

Features

- Transmit distance up to ١٠٠٠m (line of sight) @٩٦٠٠ bps
- ٢٥٦ bytes data buffer
- High sensitivity (-١١٢dbbm @٩٦٠٠ bps)
- GFSK modulation
- UART/TTL interface
- Embedded watch dog
- Size: ٣٧x١٧x٦.٥ mm

Chapter Four

٤. COPTER BASIC PRINCIPLES

٤.١ Basic Quad Copter Movement

Roll tilts the quad copter left and right by speeding up the rotors on one side and slowing them down on the other.

Pitch tilts the quad copter forward and backward the same way that roll does.

Yaw rotates the quad copter by speeding up all of the rotors spinning in on direction and slowing down all of the rotors spinning in the opposite direction.

Throttle controls the up and down axis by varying the overall speed of the rotors.

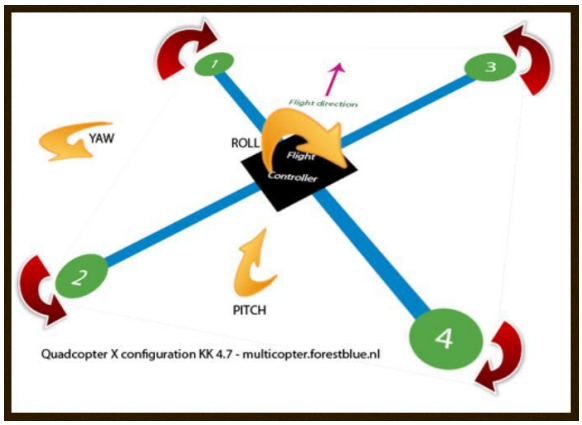


Figure ١٩ rotation angles around copter

Each rotor produces a thrust and a torque about its center of rotation and these forces are used to fly and move Quad copter. Two rotors mounted on opposite arms of quad copter are set into clockwise and another two anticlockwise. These orientations of motors and their direction of rotation cancels all the torque generated given the speed of the motors are same. Now, if we change the speed of the motors attached on the left and right of quad keeping thespeed of the other two same, creates 'Yaw' motion. Similarly, 'Pitch' and 'Roll' movements are gained by changing the speed of different motors. See the images below.

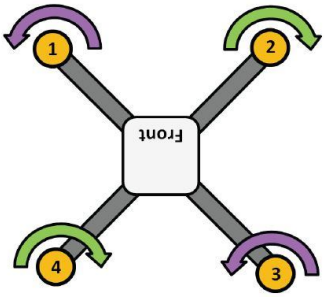
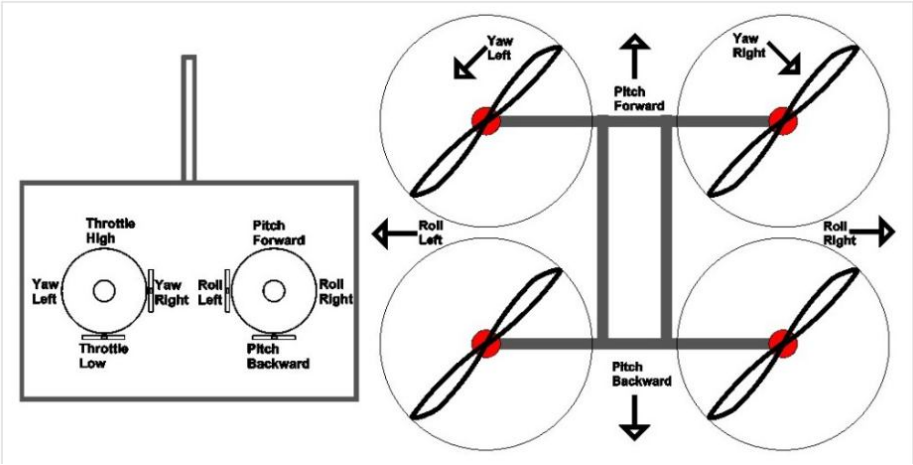


Figure ٢٠ motors rotation directions

There are four main quad copter controls:

- Roll
- Pitch
- Yaw
- Throttle



Roll

By moving, the roll stick to the right

- We increase the speed of the left motors, and vise reveres.

Here, the bottom of the propellers will be

- Facing to the left. This pushes air to the left, forcing the quad copter to fly to the right.

Figure ٢١ simple sketch of (roll, yaw, pitch and throttle) on transmitter (left image) and Quad copter (right image)

- The same thing happens when you push the stick to the left, except now the

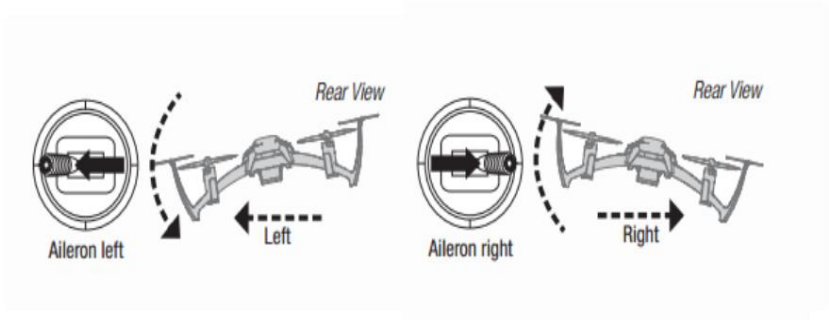


Figure ٢٢ copter motion with roll

propellers will be pushing air to the right, forcing the copter to fly to the left.

Pitch

By moving, the pitch stick to the up

- We increase the speed of the backward
- Motors and vise reveres.

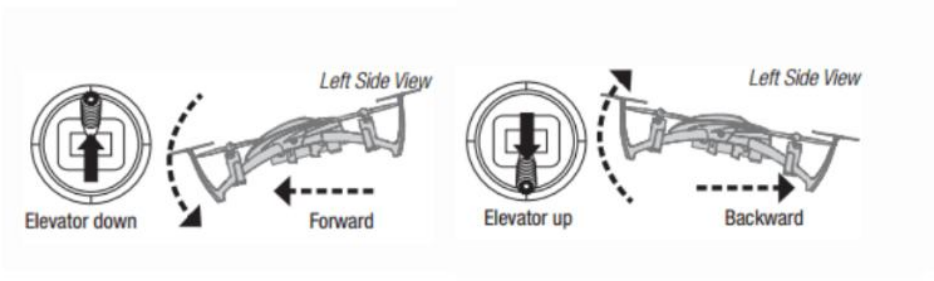


Figure ٢٣ copter motion with pitch

Yaw

By moving, the yaw stick to the right

- We increase the speed of the clockwise
- Motors so the quad copter rotate right around its center,

By moving, the yaw stick to the left

- We increase the speed of the anti-clockwise

- Motors so the quad copter rotate left around its center.

Throttle

Throttle gives the propellers onquad copter enough power to get airborne. When flying, the throttle will be engaged constantly.

For the Quad copter the fly properly two conditions must be full filled

١. Thrust force is equal or more than the weight force (gravitational force)

This is done by making the four motors and the propellers push the air into the ground.

٢. The torque around its center is equal to zero

This is done by making each two crossing motors rotors in a direction opposite to the other two motors.

٤.٢ Basic Copter Calculations

We get the average weight of the component and compare to thrust force of the four motors to calculate at which throttle the copter is going to take off.

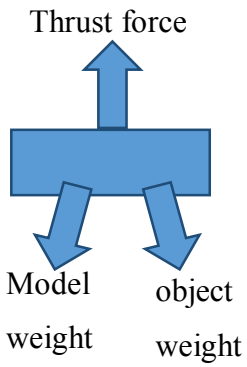


Figure ٢٦ simple illustration of forces on copter

The weight of the copter consists of the weight of total copter model components and object (according to the application) that the copter are going to carry.

Weight of the copter

Hardware name	Weight
Motors	٤ * ٦٨ = ٢٧٢
Propellers	٤ * ١٠ = ٤٠
frame	٣٥٠
ESC	٤ * ٤١ = ٨٤
Battery	٤٩٠
Power distribution board	١٥
Arduino nano	٤٠
Arduino mega	٥٠
Mobile	١٤٠
RF	٢٠
GPS	٣٦
Gyroscope	١٤
Wires and tapes	١٠٠
Ultrasonic	٢٠
Resistors and welding	٢٩
Total	١٧٠٠ gm.

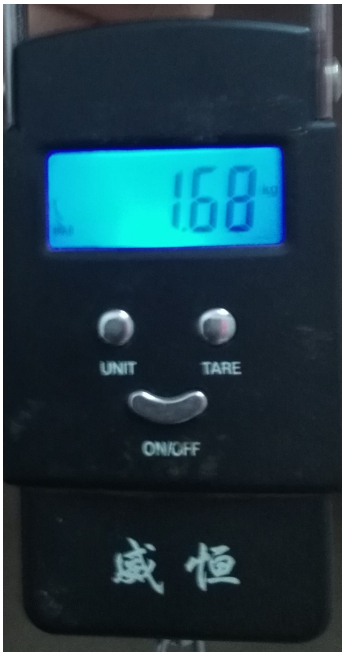


Figure ٢٧ the weight is almost ١.٧ Kg

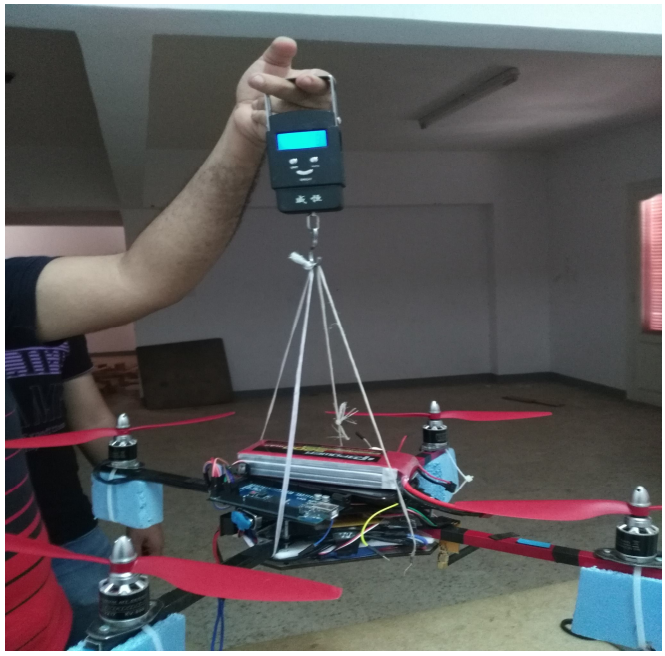


Figure ٢٨ weighting our model of the copter

After calculating the weight, we add ١٠% tolerance for better results.

Thrust force for one motor is ٨٠٠ gm

Brushless motor efficiency is ٨٠ %

The percentage of throttle that copter take off is

Giving us about ٣٠ percentage of remaining throttle to rise the copter in the air.

Factors that affect thrust force

- Propellers (pitch, diameter and material of the propellers).
- Motors (RPM)
- Atmospheric air density (normal day: ١.٢٥٥ kg/m^{٣})

Calculating the maximum RPM of one motor

$$maximum\ RPM = KV\ of\ the\ motor * battery\ voltage$$

$$maximum\ RPM = ٨٠٠ * ١١.١ = ٨٨٠٠\ rpm$$

LiPo battery Calculation

$$maximum\ discharge\ rate = battery\ capacity\ (mAh) * peak\ discharge$$

$$maximum\ discharge\ rate = ٥.٢ * ٣٥ = ١٨٢\ A$$

Maximum discharge rate is although known as maximum drawn current from battery.

$$maximum\ current\ required = motor\ maximum\ current * no\ of\ motors$$

$$maximum\ current\ required = ١٥.٤ * ٤ = ٦١.٦\ A$$

Maximum required current must be less than Maximum discharge rate.

Calculating the flight time

$$flight\ time = \frac{capacity}{average\ current\ drawn} * ٦٠$$

$$fligth\ time = \frac{٥.٢}{٠.٧ * ٤ * ١٥.٤} * ٦٠ = ٧.٢٣٦\ minute$$

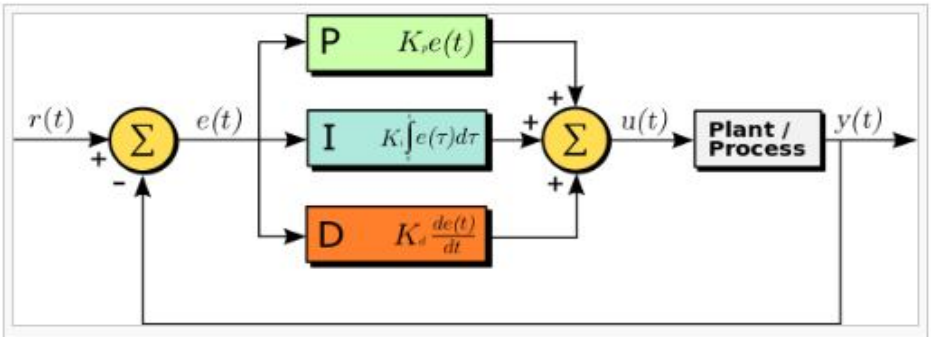
$$average\ flight\ time = ٧.٢٣٦ * ٠.٨ \approx ٦\ minutes$$

Chapter Five

٥. CONTROL

٥.١ PID Controller

A proportional–integral–derivative controller (PID controller) is a control feedback mechanism (controller) commonly used in industrial. A PID controller continuously calculates an error value as the difference between a desired set point and a measured process variable. The controller attempts to minimize the error over time by adjustment of a control variable, such as the position of a control valve, a damper, or the



power supplied to a heating element, to a new value

As a PID controller relies only on the measured process variable, not on knowledge of the underlying process, it is broadly applicable. By tuning the three parameters of the model, a PID controller can deal with specific process requirements. The response of the controller can be described in terms of its responsiveness to an error, the degree to which the system overshoots a set point, and the degree of any system oscillation. The use of the PID

Figure ٢٤ block diagram of PID feedback loop

algorithm does not guarantee optimal control of the system or even its stability.

PID controller theory

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

Where

$$K_p : proportional\ gain, a\ tuning\ parameter$$

$$K_i : integral\ gain, a\ tuning\ parameter$$

$$K_d : derivative\ gain, a\ tuning\ parameter$$

$$e(t) : \ Error = SP - PV(t)$$

$$SP : set\ point$$

$$PV(t) : process\ variable$$

$$t : time\ or\ instantaneous\ time$$

$$\tau : variable\ of\ integration$$

Proportional term

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant.

The proportional term is given by:

$$P(out) = K_p e(t)$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

The p-regulator can achieve stability but not tracking which mean that the output cannot reach the input with using of p-regulator only.

The other thing is that the p-regulator is not only robust.

Integral term

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.

The integral term is given by:

$$I(out) = k_i \int_{.}^{t} e(\tau) d\tau$$

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the set point value.

Derivative term

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d.

The derivative term is given by:

$$D(out) = k_d \frac{de}{dt}$$

Derivative action predicts system behavior and thus improves settling time and stability of the system.

Making the PID to control the Quad Copter

- The PID Set point: is the signal from the remote controller (converted into degree per second).
- The PID Feedback: is the system output measured by a Gyroscope sensor (converted into degree per second).
- The PID output: is then converted (compensated) into proper pulses for the speed controller (ESCs) to control the four motors.

The three parameter that we want to control are

- The Pitch angle (the rotation angle about the y axis)
- The Roll angle (the rotation angle about the x axis)
- The Yaw angle (the rotation angle about the z axis)

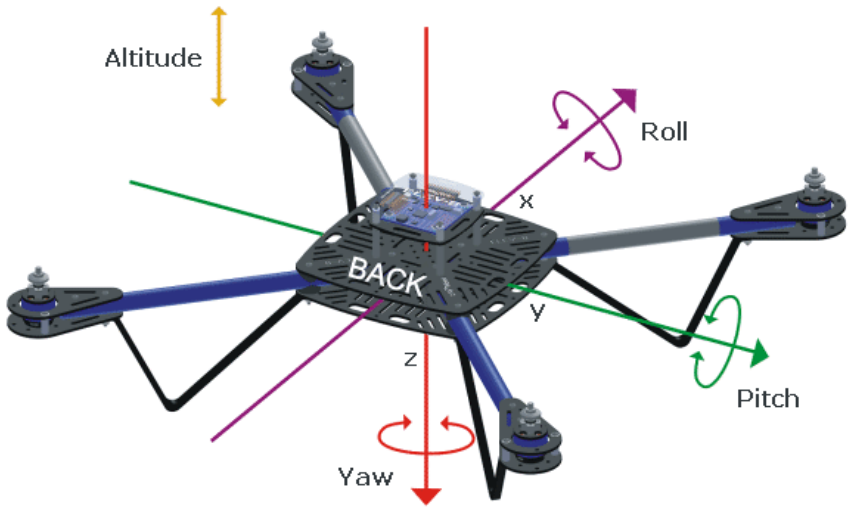


Figure٢٠ figure showing the parameters that are wanted to be controlled

The PID is done in ATmega١٣٨ (Arduino Nano). There are three PID controllers in Arduino Nano: the Roll PID, the Pitch PID and the Yaw PID. The equation for the three is the same; the difference is in the PID constant of them. However, the PID constants for Roll PID and Pitch PID is the same.

The Roll PID

The equations for roll PID controller are:

$$e_{new-roll} = r_{roll} - y_{roll}$$

$$r : reference\ (Roll\ set\ point)$$

$$y : output\ (gyroscope\ roll\ angle)$$

١. The P controller

$$P_{roll} = K_{p-roll} e_{new-roll}$$

٢. The I controller

$$I_{roll-new} = I_{roll-old} + K_{i-roll} e_{new-roll}$$

$$I_{roll-old} = I_{roll-new}$$

٣. The D controller

$$D_{roll} = K_{d-roll}(e_{new-roll} - e_{old-roll})$$

$$e_{old-roll} = e_{new-roll}$$

The Roll PID output

$$u_{roll} = P_{roll} + I_{roll} + D_{roll}$$

The Roll PID constant after a few try and error are

$$K_{p-roll} = ١.٤$$

$$K_{i-roll} = ٠.٠٥$$

$$K_{d-roll} = ١٥$$

The Pitch PID

The equations for pitch PID controller are:

$$e_{new-pitch} = r_{pitch} - y_{roll}$$

$$r : reference\ (pitch\ set\ point)$$

$$y : output\ (gyroscope\ pitch\ angle)$$

١. The P controller

$$P_{pitch} = K_{p-pitch} e_{new-pitch}$$

٢. The I controller

$$I_{pitch-new} = I_{pitch-old} + K_{i-pitch} e_{new-pitch}$$

$$I_{pitch-old} = I_{pitch-new}$$

٣. The D controller

$$D_{pitch} = K_{d-pitch}(e_{new-pitch} - e_{old-pitch})$$

$$e_{old-pitch} = e_{new-pitch}$$

The pitch PID output

$$u_{pitch} = P_{pitch} + I_{pitch} + D_{pitch}$$

The Pitch PID constant after a few try and error are

$$K_{p-Pitch} = ١.٤$$

$$K_{i-pitch} = ٠.٠٥$$

$$K_{d-pitch} = ١٥$$

The Yaw PID

The equations for yaw PID controller are:

$$e_{new-yaw} = r_{yaw} - y_{yaw}$$

$$r : reference\ (Yaw\ set\ point)$$

$$y : output\ (gyroscope\ yaw\ angle)$$

١. The P controller

$$P_{yaw} = K_{p-yaw} e_{new-yaw}$$

٢. The I controller

$$I_{yaw-new} = I_{yaw-old} + K_{i-yaw} e_{new-yaw}$$

$$I_{yaw-old} = I_{yaw-new}$$

٣. The D controller

$$D_{yaw} = K_{d-yaw}(e_{new-yaw} - e_{old-yaw})$$

$$e_{old-yaw} = e_{new-yaw}$$

The Yaw PID output

$$u_{yaw} = P_{yaw} + I_{yaw} + D_{yaw}$$

The yaw PID constant after a few try and error are

$$K_{p-yaw} = ٤.٠$$

$$K_{i-yaw} = ٠.٠٢$$

$$K_{d-yaw} = ٠.٠$$

After the PID output is calculated then these values are used to calculate the pulse width input to the ESCs to give proper movement of the quad copter.

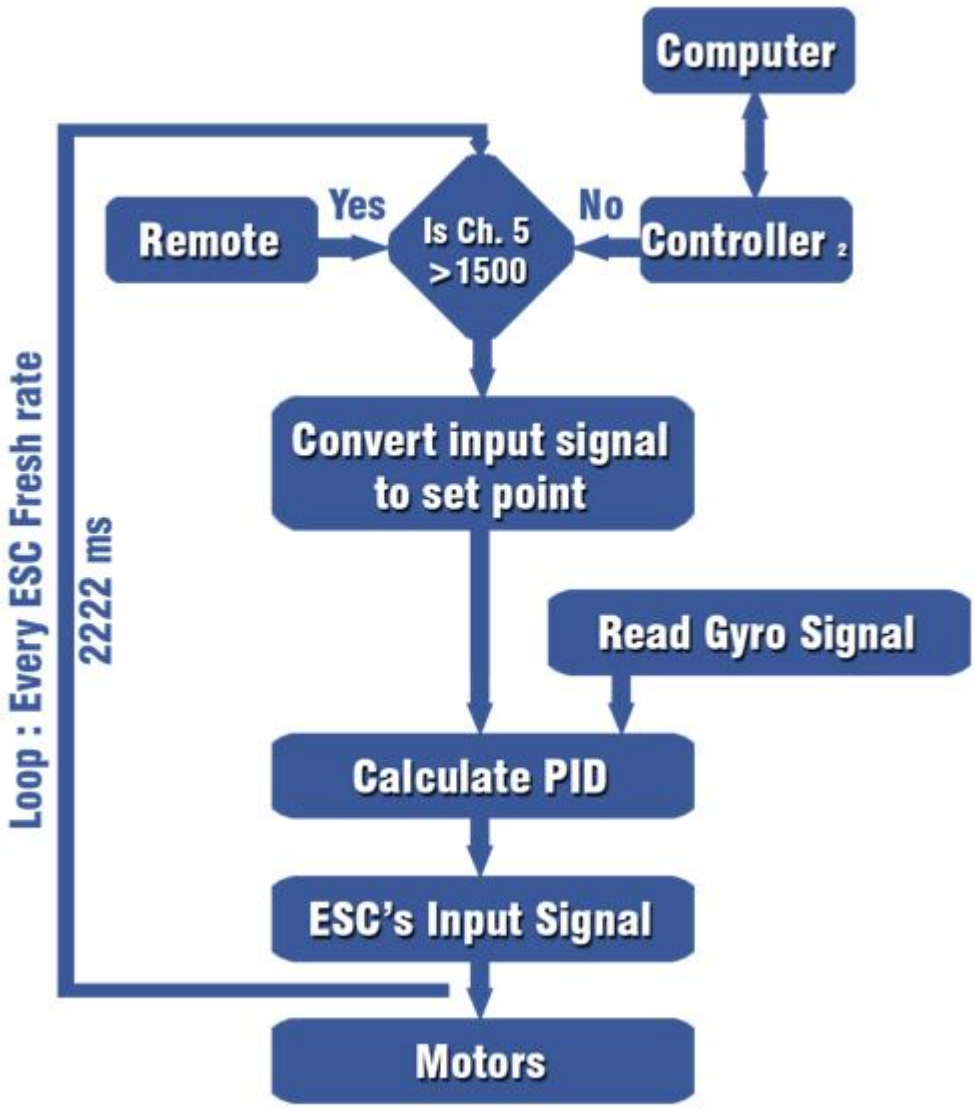


Figure ٢٥ PID program flow

The ESCs PWM Pulse Width equations

١. $ESC_{١}$ (for front right motor moving counter-clockwise)

$$ESC_{١} = Throttle - u_{pitch} + u_{roll} - u_{yaw}$$

٢. $ESC_{٢}$ (for rear right motor moving clockwise)

$$ESC_{٢} = Throttle + u_{pitch} + u_{roll} + u_{yaw}$$

٣. $ESC_{٣}$ (for rear left motor moving counter-clockwise)

$$ESC_{٣} = Throttle + u_{pitch} - u_{roll} - u_{yaw}$$

٤. $ESC_{٤}$ (for front left motor moving clockwise)

$$ESC_{٤} = Throttle - u_{pitch} - u_{roll} + u_{yaw}$$

Then these values are continuously used to change the ESCs PWM Pulse width inputs.

٥.٢ Controller

Another control part is done by Arduino Mega. We choose Arduino mega because it has strong features like four USART that allowed us to connect various parts on the same controller, and five timers to generate PWM signal wanted.

The mega controller is connected to

١. The computer through RF transceiver (UART١)
٢. The PID controller
٣. Mobile phone through UART٠
٤. GPS through UART٢
٥. Ultrasonic Range detection sensor

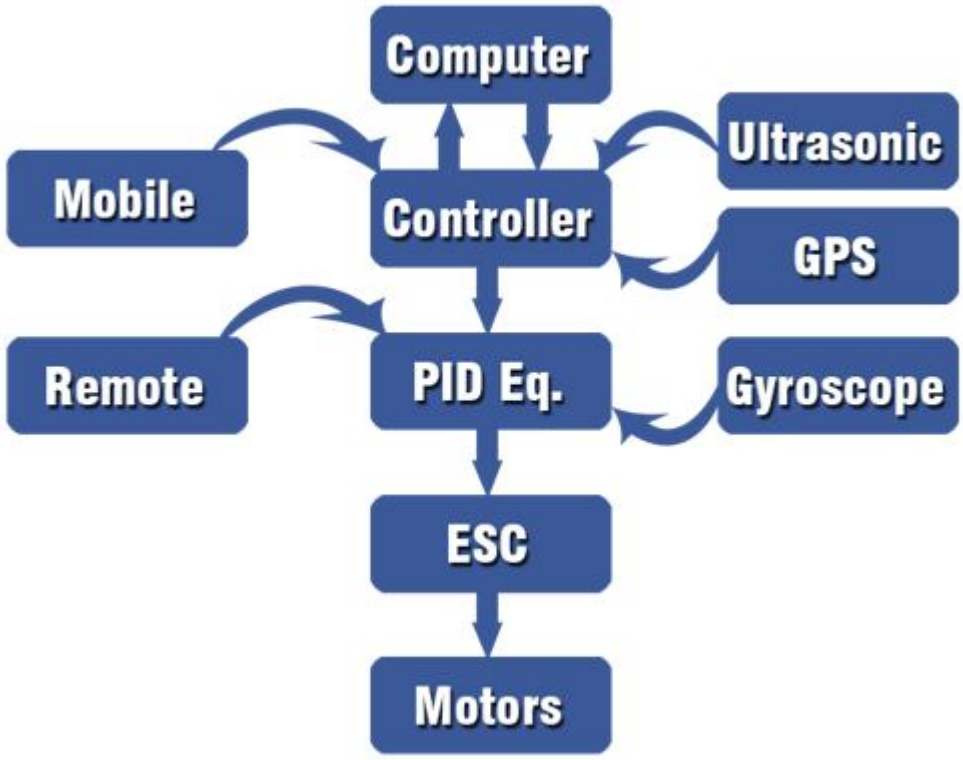


Figure ٢٦ project layout

Through programming this controller, we are able to manage the communication between project's parts, receiving commands from the computer, continuously sending feedback data to the computer from the data received from the mobile (application part of the project), GPS, Ultrasonic and information about the flight status and health. Although the controller is able to translate the commands from into proper PWM signal (like the RC remote receiver's PWM signal) to the PID controller making it able to control the copter's flight just like the remote.

Controller connection with the computer

The computer and the controller are connected by the APP٢٢٠ wireless kit, which is consisting of two transceivers and USB TTL. The computer is connected to USB TTL then connected to the APC٢٢٠ transceiver. The controller is connected to the other APC٢٢٠ transceiver. The APC٢٢٠ can be configured through "RF magic" configuring tool to the wanted baud rate,number of bits in frame, operating RF frequency and stopping and starting bits. The communication between the computer and the controller is done through UART communication protocol.

Predefined characters must label any sending data between computer and controller. The next table show the data sent from (computer and controller) and the meaning of it and their predefined char label.

Data from computer

Data or Command	Data form	Label character	Notes
Start copter	‘S’	‘S’	Here the label is the same as data, so when the controller receives upper case ‘S’, it starts the copter.
Land copter	‘L’	‘L’	Here the label is the same as data, so when the controller receives upper case ‘L’, it lands the copter.
Stop copter	‘P’	‘P’	Here the label is the same as data, so when the controller receives upper case ‘P’, it stops the copter.
Start sending data	‘T’	‘T’	Here the label is the same as data, so when the controller receives upper case ‘T’, the controller starts send data to the computer.
Stop sending data	‘N’	‘N’	Here the label is the same as data, so when the controller receives upper case ‘N’, the controller stops send data to the computer.
Throttle value	Value received char by char like ‘١’ ‘٢’ ‘٣’	‘t’	When the controller receives lower case ‘t’, then it starts receiving integer characters until the non-integer character.
Pitch value	Value received char by char like ‘١’ ‘٢’ ‘٣’	‘p’	When the controller receives lower case ‘p’, then it starts receiving integer characters until the non-integer character.
Roll value	Value received char by char like ‘١’ ‘٢’ ‘٣’	‘r’	When the controller receives lower case ‘r’, then it starts receiving integer characters until the non-integer character.
Yaw value	Value received char by char like ‘١’ ‘٢’ ‘٣’	‘y’	When the controller receives lower case ‘y’, then it starts receiving integer characters until the non-integer character.
Indicator about finishing the sending of controllers values	‘C’	‘C’	Here the label is the same as data, so when the controller receives upper case ‘C’, the controller indicates that controller values has been sent.

Data to computer

a. Data about batteries health

Data	Data form	Label character	Notes
Motor's battery	Value send char by char like '١' '٢' '٣'	'H'	
PID controller's battery	Value send char by char like '١' '٢' '٣'	'B'	
Arduino Mega battery	Value send char by char like '١' '٢' '٣'	'F'	

b. Data from GPS

Data	Data form	Label character	Notes
Latitude	Value send char by char like '١' '٢' '٣'	'G'	
Longitude	Value send char by char like '١' '٢' '٣'	'D'	
Speed	Value send char by char like '١' '٢' '٣'	'M'	
Number of satellite seen by GPS	Value send char by char like '١' '٢' '٣'	'N'	
Altitude above sea level	Value send char by char like '١' '٢' '٣'	'A'	

c. Ultra sonic range (height) detection

Data	Data form	Label character	Notes
Height from ultra-sonic	Value send char by char like '١' '٢' '٣'	'K'	

d. Mobile drive test data (application)

Data	Data form	Label character	Notes
Current signal strength	Value send char by char like '١' '٢' '٣'	'a'	
Signal level	Value send char by char like '١' '٢' '٣'	'b'	
Mobile cell ID	Value send char by char like '١' '٢' '٣'	'c'	
Location Area	Value send char by char like '١' '٢' '٣'	'd'	
Operator name	Value send char by char like '١' '٢' '٣'	'o'	
MCC mobile	Value send char by	'f'	

country code	char like '١' '٢' '٣'		
Data	Data form	Label character	Notes
SIM state	Value send char by char like '١' '٢' '٣'	'g'	
Roaming state	Value send char by char like '١' '٢' '٣'	'i'	
Network type	Value send char by char like '١' '٢' '٣'	'h'	
Phone network type	Value send char by char like '١' '٢' '٣'	'j'	
IMEI	Value send char by char like '١' '٢' '٣'	'k'	
IMSI	Value send char by char like '١' '٢' '٣'	'l'	
Voice capability	Value send char by char like '١' '٢' '٣'	'm'	
SIM serial	Value send char by char like '١' '٢' '٣'	'n'	

Controller connection with PID controller

The motors are controlled by ESCs (electronic speed controller). The ESCs are controlled by PWM signal (which is originally generated by the RC remote) with frequency defined by the ESC type, but in general, the ESC gives the wanted power to the motor according to the width of the PWM signal which varies from ١٠٠٠ microsecond to ٢٠٠٠ microsecond pulse width corresponding to minimum to maximum power respectively. The PID controller take the RC remote signal (the control signal of the ESC) as an input to PID calculation with the gyroscope feedback sensor to generate the ESC control pulses to perform the proper (wanted) movement of the four motors.

To make the computer (GUI) control the flight of the copter as the RC remote and Receiver, the controller has to translate the flight commands from the GUI into PWM signals with the proper width (just like the RC receiver) as inputs to PID controller.

This is done by using the timers of ATmega٢٥٦٠ (Arduino mega) to generate the PWM signals with wanted frequency and continuously changing period. The controller generate four PWM signals for the main four controller parameters (throttle, pitch, roll, and yaw). These four signal is then used as input to the PID controller as the RC remote Receiver signal. With the proper switch you can choose to control the copter with the RC remote and Receiver or with the GUI and the ATmega٢٥٦٠ controller.

The mathematics for transferring from the received data from the GUI to the proper PWM signal width is quite easy.

The control data parameters are received as integer value from ٠ to ٢٥٥ corresponding to ٠ to ١٠٠٠ respectively. In addition, with simple mapping these values is transformed to a value from ٠ to ١٠٠٠, and then is added to ١٠٠٠ to get the wanted PWM pulse width for each of the four control parameters.

$$PWM\ pulse\ width = \left(\frac{١٠٠٠}{٢٥٥}\right) * received\ control\ value$$

Controller connection with GPS

Our Quad Copter application is Drive Test, which is testing the mobile network signal over many places. The GPS was the solution for this problem. The GPS give us the data into NEMA form and we use “tiny_gps++” library to transform the data from this form into readable data. Therefore, the controller continuously takes GPS data and sends them to the GUI for location update and database build.

The data that is got from the GPS is like

- Latitude
- Longitude
- Speed
- Number of satellites seen by GPS
- Altitude above sea leve

Controller connection with Ultra sonic

We use the ultra-sonic range detection to give us the data about the actual height of the quad copter. This is done by placing one ultra-sonic under the copter. This gives us the height range from ٢ cm to ٥ meters in centimeters resolution. Then is data is although sent to the GUI for continuous update.

Controller connection with Mobile Phone

The mobile does the application of the copter. We made an android application that extracts the wanted data from the mobile device. Then the mobile is connected to the controller through the OTG (on the go) cable allowing the mobile to send the data to the controller

Figure ٢٧ OTG cable

through UART communication protocol.

Mobile drive test's data (application) send to Arduino Mega

Data	Data form	Label character	Notes
Current signal strength	Value send char by char like '١' '٢' '٣'	'a'	
Signal level	Value send char by char like '١' '٢' '٣'	'b'	
Mobile cell ID	Value send char by char like '١' '٢' '٣'	'c'	
Location Area	Value send char by char like '١' '٢' '٣'	'd'	
Operator name	Value send char by char like '١' '٢' '٣'	'%'	
MCC mobile country code	Value send char by char like '١' '٢' '٣'	'f'	
SIM state	Value send char by char like '١' '٢' '٣'	'g'	
Roaming state	Value send char by char like '١' '٢' '٣'	'i'	
Network type	Value send char by char like '١' '٢' '٣'	'h'	
Phone network type	Value send char by char like '١' '٢' '٣'	'j'	
IMEI	Value send char by char like '١' '٢' '٣'	'k'	
IMSI	Value send char by char like '١' '٢' '٣'	'l'	
Voice capability	Value send char by char like '١' '٢' '٣'	'm'	

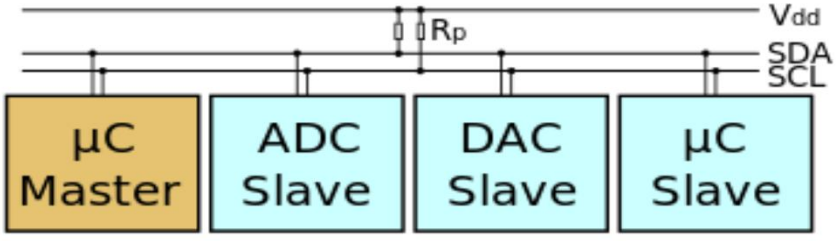
SIM serial	Value send char by char like '١' '٢' '٣'	'n'	

Chapter Six

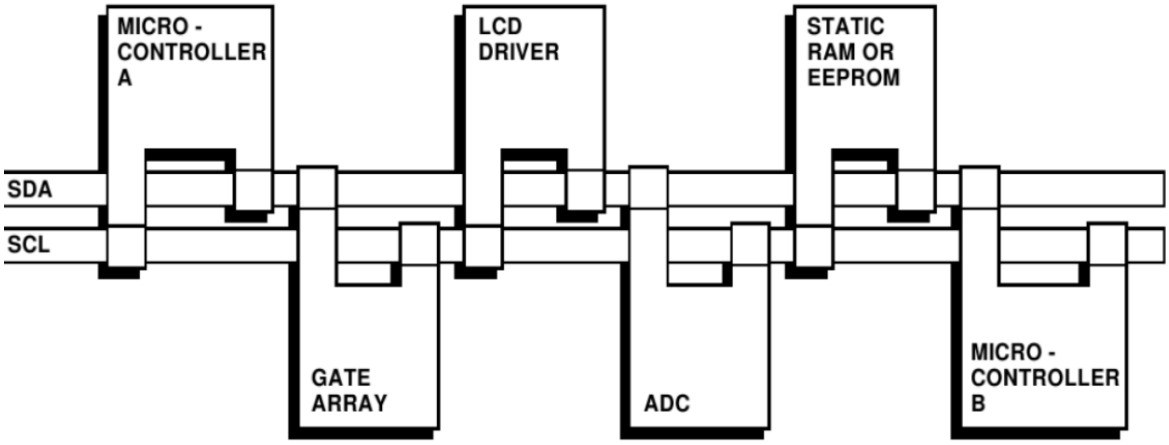
٦. COMMUNICATION BETWEEN CONTROLLERS

٦.١ Inter Integrated Circuit(I٢C)

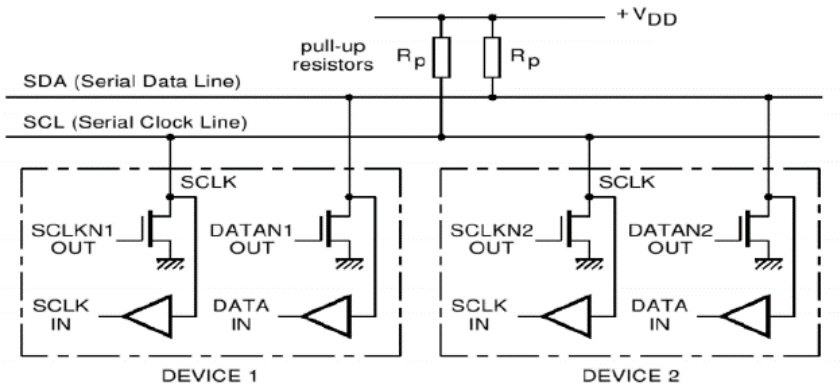
- Inter Integrated Circuit (I٢C) and called Two Wire Interface (TWI).
  I٢C/TWI is widely used interface in embedded applications. Two-wire bus initially was used by Philipsand become a standard among chip vendors.
- I٢C/TWI is a half-duplex serial transmission so the data flow can be in one direction at a time.
- I٢C/TWI is Synchronous communications.
- Multi-Master Multi-Slave Bus.
- ٧-bit address space Maximum ١١٢ nodes (١٦ reserved).
  All devices connected to the bus have unique address.
- Speed Modes:
  Normal Mode ١٠٠ Kb/s (The most common).
  Fast Mode ٤٠٠ kb/s.
  High-Speed Mode ٣.٤ Mb/s.
- The ACK (acknowledgement) signal is sent/received from both the sides after every transfer and hencereduces the error.
- Widely used in fast and short distance communicationlike communicate between sensors, EEPROM and LCD.
- The two I٢C signals are serial data (SDA) and serialclock (SCL). Together, these signals make it possible tosupport serial transmission of ٨-bit bytes of data and ٧-bit device addresses and control bits-over the two-wireserial bus. The device that initiates a transaction on theI٢C bus is termed the master. The master normally



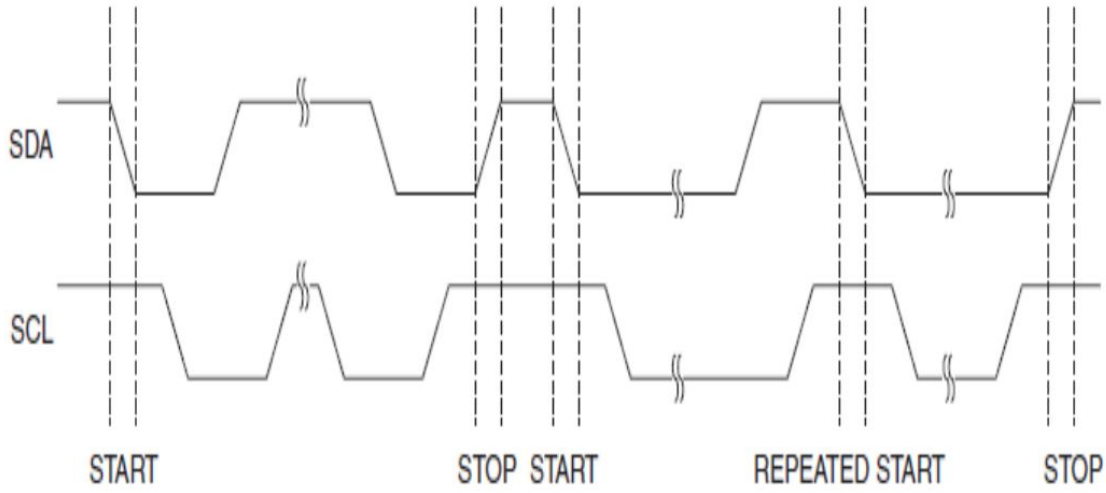
controls the clock signal. A device being addressed bythe master is called a slave.



- Open-drain (also called open-collector) lines pulled upwith resistors. What does that mean? It means that thedevices connected to the I²C bus are capable of pullingany of these two lines low, but they cannot drive themhigh. If any of the devices would ever want to drive thelines high, they would simply need to let go of that line,and it would be driven high by the pull up resistors.



Each communication is initiated by START signal andfinished by STOP. Master always generates these. START and STOP signals are generated bypulling SDA line low and pulling SDA line high while SCLline is high.



I٢C protocol Sequence

١. Master initiate the clock.

٢. Master sending a start bit (S= '٠').

٣. The master sending a unique ٧-bit slave deviceaddress.

٤. Then send a read/not-write bit:

- '١' Read from Slave.
- '٠' write to Slave.

٤. This is followed by an ACK bit ('٠') send by the slave.

٥. Then the transmitter (slave or master, as indicated bythe read/not-write bit) transmits a byte of data startingwith the MSB first.

٦. This is followed by an ACK bit ('٠') send by the receiver.

٧. This ٩-bits pattern (Data + ACK) is repeated if morebytes need to be transmitted.

٨. At end master send stop bit ('١')

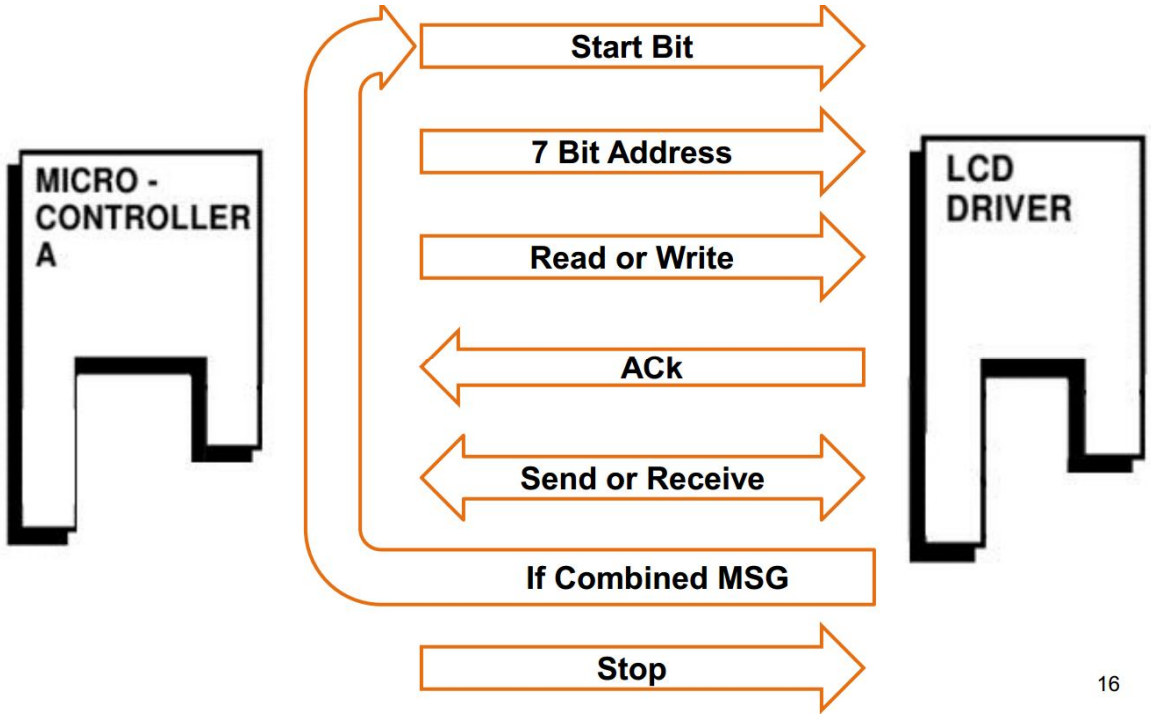


Figure ٣٤ I٢C communication protocol

٦.٢ Universal asynchronous receiver/transmitter (USART)

USART stands for Universal Synchronous Asynchronous Receiver Transmitter. It issometimes called the Serial Communications Interface or SCI. Both transmission and reception can occur at the same time, this is known as full duplex operation. Datacommunication can be achieved by two methods: Synchronous and Asynchronous communication. There are two data lines, RX and TX, which stand for receiver and transmitter, relative to a device.

Data transmission:

In Synchronous communication method: one device is configured as the MASTER andgenerates the clock and the other is configured as the SLAVE. The MASTER cantransmit data at any time by generating clock signal but the SLAVE must wait theclock signal to end data. Complete byte (character) is sent at directly on the clockedges and doesn't require any additional bits to be added for the synchronization asthe devices are synchronized by clock.

In asynchronous communication method: there's no MASTER or SLAVE and eachdevice generates its own clock, so both devices can transmit data at any time. Thebyte (character) is transmitted in a frame and additional bits are added for thesynchronization of frame start, stop or parity).

Applications:

The USART is most commonly used in the asynchronous mode. The most commonuse ofthe USART in asynchronous mode is to communicate to a PC serial port using٧٨the RS-٢٣٢ protocol.

In normal microcontroller circuits the TTL (Transistor-TransistorLogic) standard for serial communication is used, where High = +٥V, Low = ٠V.But computers use RS٢٣٢ communication standard, the protocol which specifies High= -١٢V, Low = +١٢V. A driver is required to interface to RS-٢٣٢ voltage levels. Thisconversion is achieved through a MAX٢٣٢ integrated circuit (the microcontrollershould not be directly connected to RS-٢٣٢ signals).

Chapter Seven

٧. APPLICATION “DRIVE TEST” and ANDROID APPLICATION

٧.١ Drive Test

Drive testing is a method of measuring and assessing the coverage, capacity and Quality of Service (QoS) of a mobile radio network.

The technique consists of using a motor vehicle containing mobile radio network air interface measurement equipment that can detect and record a wide variety of the physical and virtual parameters of mobile cellular service in a given geographical area.

By measuring what a wireless network subscriber would experience in any specific area, wireless carriers can make directed changes to their networks that provide better coverage and service to their customers.

Drive testing requires a mobile vehicle outfitted with drive testing measurement equipment. The equipment are usually highly specialized electronic devices. This ensures measurements are realistic and comparable to actual user experiences.

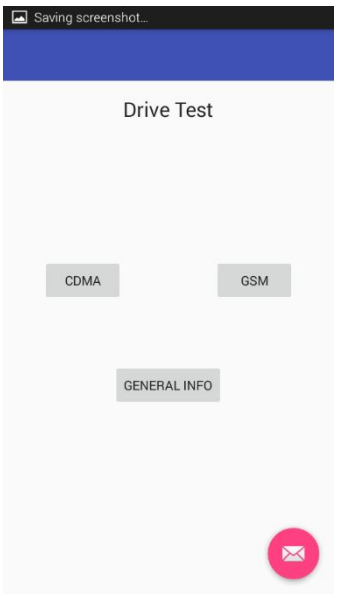


Figure ٢٨ android app main activity

٧.٢ Parameters calculated:

- Mobile network code (operatoor name): is used to identify the operator of the subscriber:

١. (٠٢) Orange.

٢. (٠١) Vodafone.

٣. (٠٣) Etisalat.

- Mobile Country Code (MCC): is used in wireless telephone networks (GSM, CDMA, UMTS, etc.) in order to identify the country which a mobile subscriber

740	02	ec	Ecuador	593	Alegro/Telcsa
740	00	ec	Ecuador	593	MOVISTAR/OteCel
740	01	ec	Ecuador	593	Porta/Conecel
602	01	eg	Egypt	20	EMS - Mobinil
602	03	eg	Egypt	20	ETISALAT
602	02	eg	Egypt	20	Vodafone/Mirsfone
706	01	sv	El Salvador	503	CLARO/CTE
706	02	sv	El Salvador	503	Digicel
706	05	sv	El Salvador	503	INTELFON SA de CV
706	04	sv	El Salvador	503	Telefonica

Figure ٢٩ countries' mobile code

belongs to , (٦٠٢) is Egypt code.

- SIM state: unknown SIM – No SIM card – PIN required – PUK required – Network locked – Ready.
- Roaming: Roaming enables a mobile subscriber to automatically make and receive voice calls, send and receive data, or access other services when travelling outside the geographical coverage area of their home network, by means of using a visited network.
- Network type: the network which the SIM using now, Ex: EDGE,CDMA,LTE …
- Phone type: indicates the type of radio used to transmit voice calls, Ex: GSM.
- International Mobile Station Equipment Identity (IMEI): the IMEI number is used by a GSM network to identify valid devices and therefore can be used for stopping a stolen phone from accessing that network.it is usually presented as a ١٥-digit. It is usually found printed on the phones back under the battery. It can also be displayed on the phones screen by entering *#٠٦# on the phones keypad. This number has three parts:
    a. Type Allocation Code (TAC): ٢ digit (country approval) + ٤ digit (phone model).
    b. Type approval code (FAC): ٢ digit (the company that had built and assembled the device).
    c. Serial number (SN): ٦ digit.
- International Mobile Subscriber Identity (IMSI): the IMSI number is used to identify the user of a cellular network and is a unique identification associated with all cellular networks. It is usually presented as a ١٥-digit. This number has two parts. The initial part is comprised five digits in the European standard, it identifies the GSM network operator (Mobile network code MNC) in a specific country (Mobile country code MCC) with whom the subscriber holds an account. The second part is allocated by the network operator to uniquely identify the subscriber.
- Voice capable: show if this device supports circuit-switched (i.e. voice) phone calls over the telephone network, or it is "data only" devices which can't make voice calls.

- SIM serial number: A SIM serial number is a unique identification number assigned to each SIM card. It is a ١٩-digit number found on the back of the card. You’ll need this for the activation process.

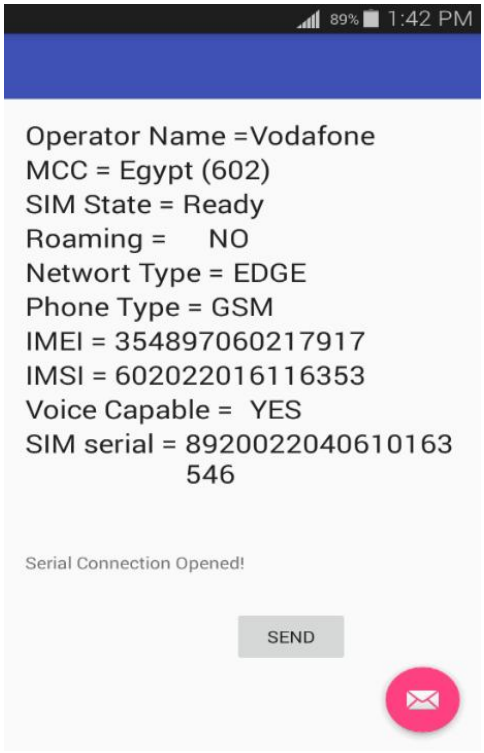


Figure ٣٠ android app general info activity

- Received Signal strength: is a measurement of the power present in a received radiosignal.

Ranges	Color	Grade
-10 > -65		Excellent
-65 > -75		Very good
-75 > -85		Good
-85 > -95		Accepted
-95 > -115		Bad

Figure ٣٨ color code for different signal strengths

- Level: a single integer from ٠ to ٥ representing the general signal quality. This may take into account many different radio technology inputs. ٠ represents very poor signal strength while ٥ represents a very strong signal strength.

/Parameters calculated:

- Cell id: A unique number used to identify the GSM base transceiver station the phone is connected to.
- Location area code (LAC): Location Area Code is a unique number of current location area. A location area is a set of base stations that are grouped together to optimize signaling.

٧.٣ How to use the application

١. Make sure the mobile device supports the OTG connection “USB host”.

٢. Connect the OTG cable with the mobile device and the Controller.

٣. Give permission to the application to access USB device.

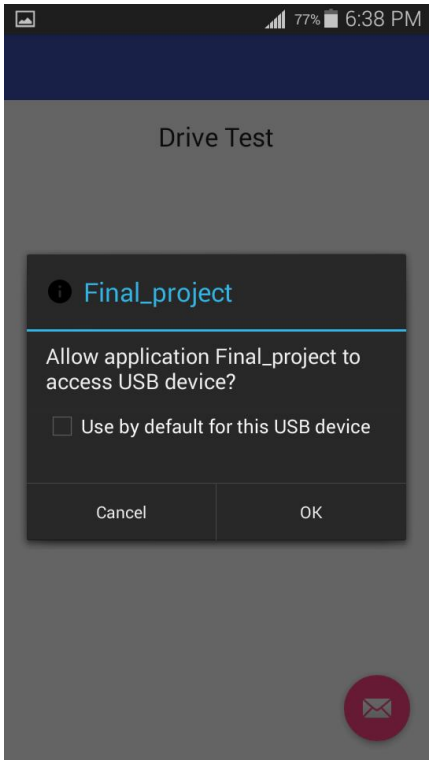


Figure ٣٩ permission for using serial

٤. Then the application is programmed to run automatically.

After that

١. The application begins to send General info through Serial port to the controller which in turn sends it to the Computer’s GUI.

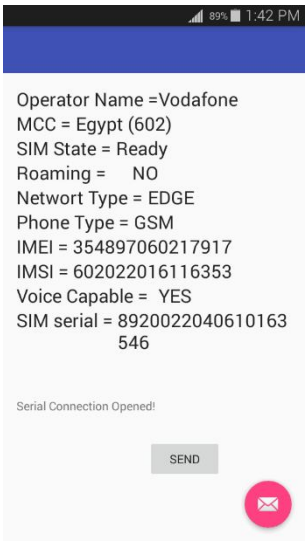


Figure ٣١ general info got from app

٢. The activity is automatically changed to GSM activity that sends the GSM info continuously and updates the Database.

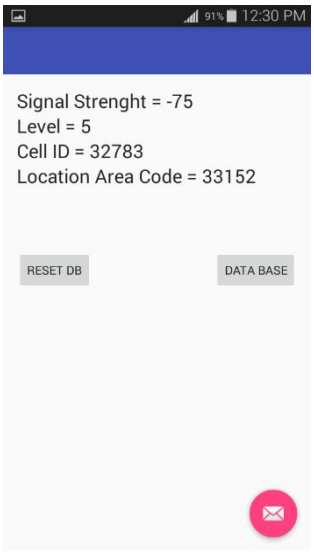


Figure ٣٢ GSM activity

Through the application you can use buttons like

١. “RESET DB” which is used to reset the database of the application.

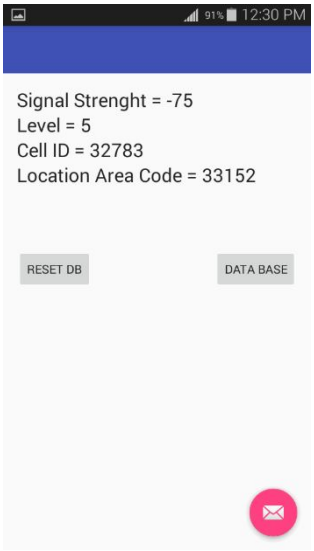


Figure ٢٤ RESET and DATA BASE buttons

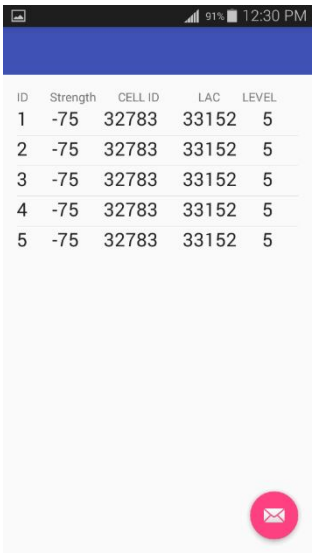


Figure ٣٣ database activity

٢. “DATA BASE” which is used to show the updated database.

Chapter Eight

٨. USER INTERFACE “GUI”

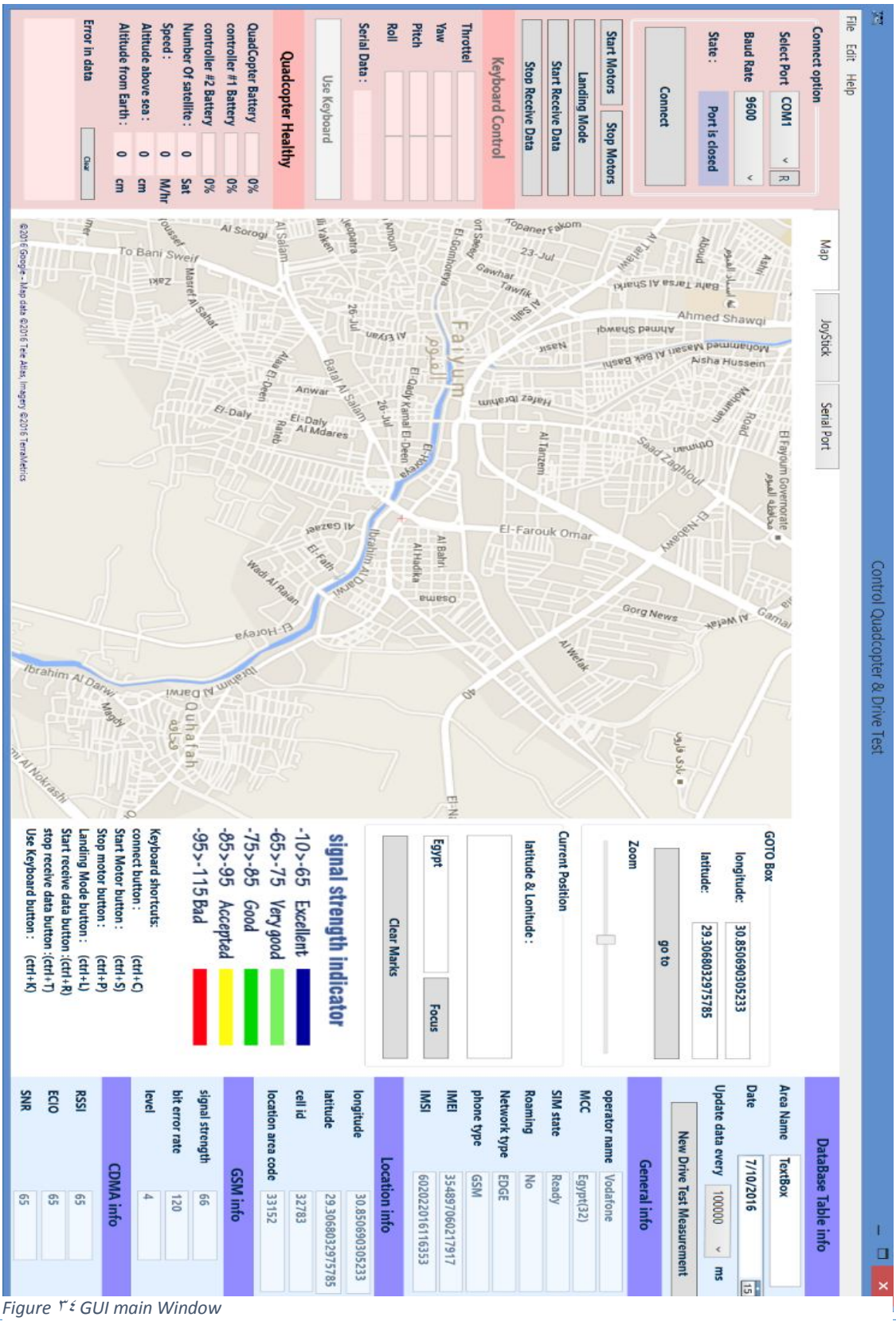


Figure ٣٤ GUI main Window

Our GUI is a ground station application for the quad copter project.

This GUI is done using C# windows presentation foundation (WPF) Language. We use WPF because it provide more facilities in graphics, maps and serial port communication.

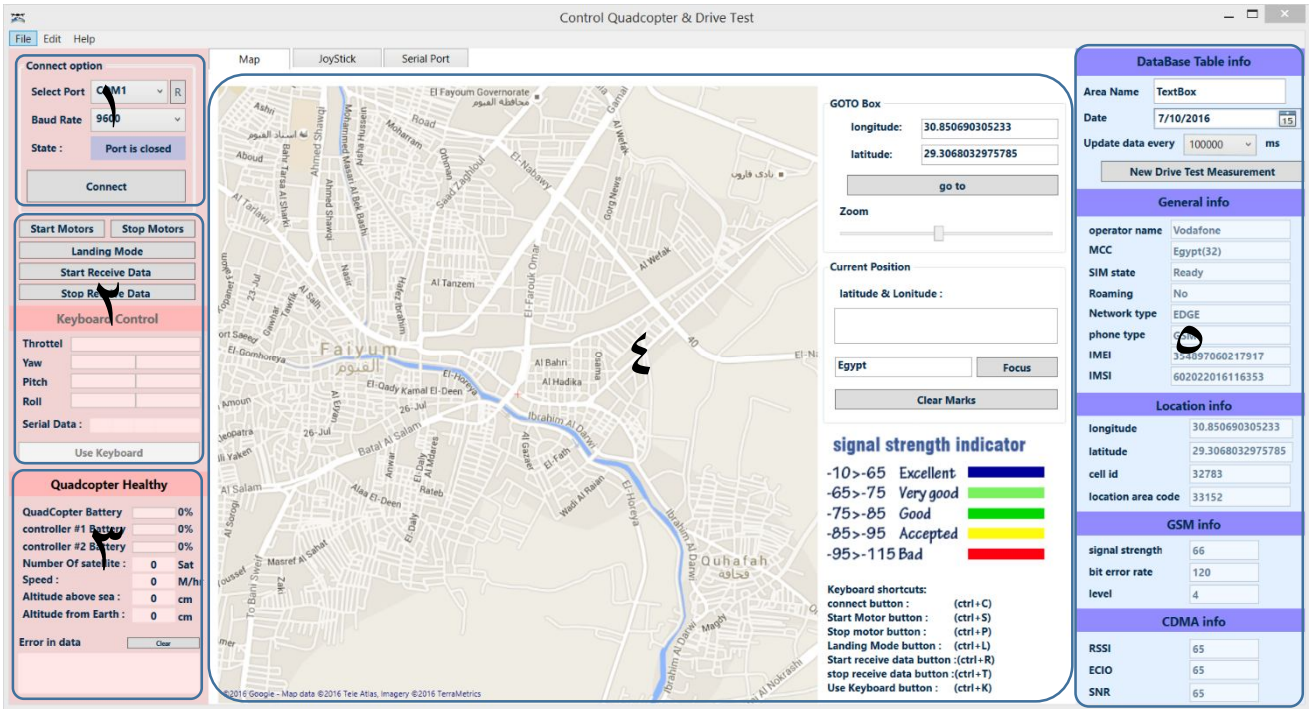


Figure ٣٥ GUI main areas

The GUI consist of many areas:

Main Window

١. Area num. ١: it’s the area where you can use to start new serial connection and select the baud rate and com port

٢. Area num. ٢: it’s the area where you can control you copter by start, stop motors and it also it have a responsibility of starting down link data

٣. Area num. ٣: throw it you can monitor the health of the copter and the data from GPS.

٤. Area num. ٤: it contain Google map provider. You can see the position of your copter on map with indicator to the signal strength.

٥. Area num. ٥: throw it you can monitor the all the data which the mobile measure.

٦. Area num. ٦: joystick area use to control your copter with joystick

٧. Area num. ٢: serial port area use if you want to send any command

Now, we start to get more information about each area

٨.١ Area num. ١ : Serial port connection

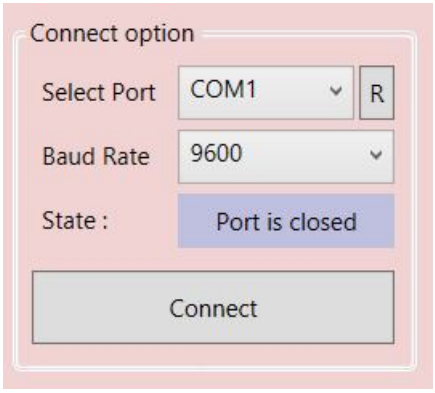


Figure ٣٦ area ١ for starting connection

Object	Object Name	Action	notes
Buttons			
Connect	btnConnect	Start serial Port communication	Ctrl +C
R	btnRefresh	Refresh serial port com	
Combo box			
Select port	CboxCom	Contain all connected com port	
Baud Rate	CboxBaudRate	Contain all baud rate of serial. You should select the baud rate of your RF.	
Label			
state	lblState	Show the state of serial	

٨.٢ Area num. ٢ : Motor Control

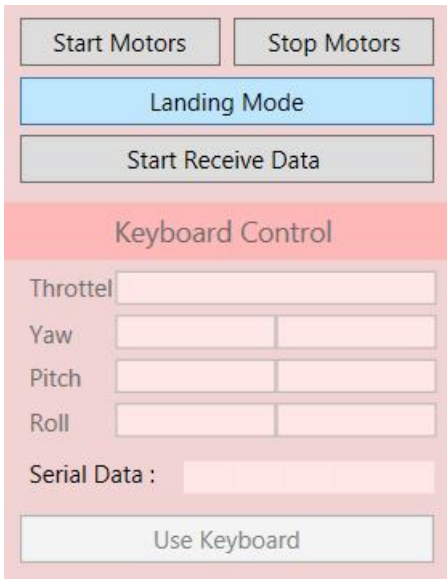


Figure ٣٧ area ٢ for controlling the copter

Object	Object Name	Action	notes
Buttons			
Start Motors	btnStartMotor	Sent “S” Command to serial port to Start Motor	Ctrl +S
Stop Motors	btnStopMotor	Sent “P” Command to serial port to Stop Motor	Cannot stopped if the throttle more than ٤٠%
Landing Mode	btnLandingMode	Sent “L” Command to serial port to Land QC	Ctrl +L
Start receive data	btnStartReceiveData	Sent “T” Command to serial port to start downlink	Ctrl +R
Start receive data	btnStartReceiveData	Sent “N” Command to serial port to start downlink	Ctrl +T
Use keyboard	btnKeyboardSelect	Allow to change angles values	Ctrl +K
Progress Bar			
throttle	pBThrottel	Indicate to the value of the throttle value	+/-
Yaw	pBYawRight	Indicate to the value of the Yaw value	٧/٩
Pitch	pBPitchRight	Indicate to the value of the Pitch value	٢/٨
Roll	pBRollRight	Indicate to the value of the Roll value	٤/٦
Label			
Serial Data	tbfirstchar	Show the value of angles which the GUI sent to copter	

٨.٣ Area num. ٣ : Data Monitoring

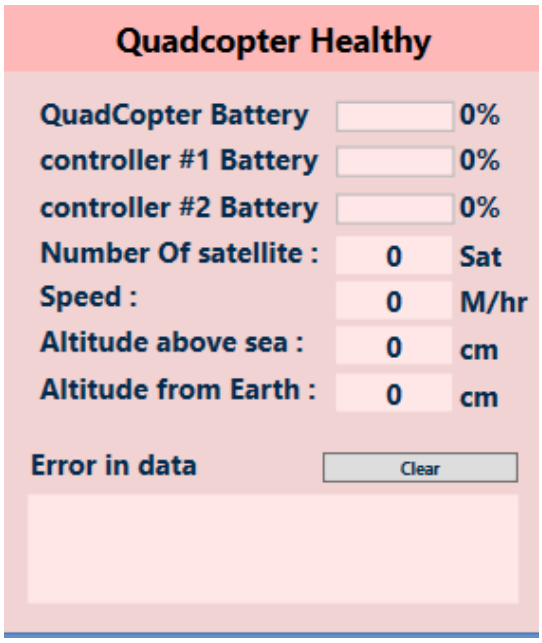


Figure ٣٨ area ٣ for monitoring the copter

Object	Object Name	Action	notes
Buttons			
Clear	btnClearErrortxt	Clear data in textbox of error	
Progress Bar			
QuadCopter Battery	pBQuadBattery	Indicate to the main battery level	
controller #١ Battery	pBController١Battery	Indicate to the value of battery of controller ١	
Text block			
Number Of satellite	txtSatNum	Number of satellite seen by GPS	
Speed	txtSatSpeed	Speed of copter measure from GPS	
Altitude above sea	txtSatAltitude	Altitude above sea of copter measure from GPS	
Altitude from Earth	txtulterasonic	Altitude from Earth of copter measure from ulterasonic	

Error in data	txtError	Any error in data	

٨.٤ Area num. ٤ : Map provider

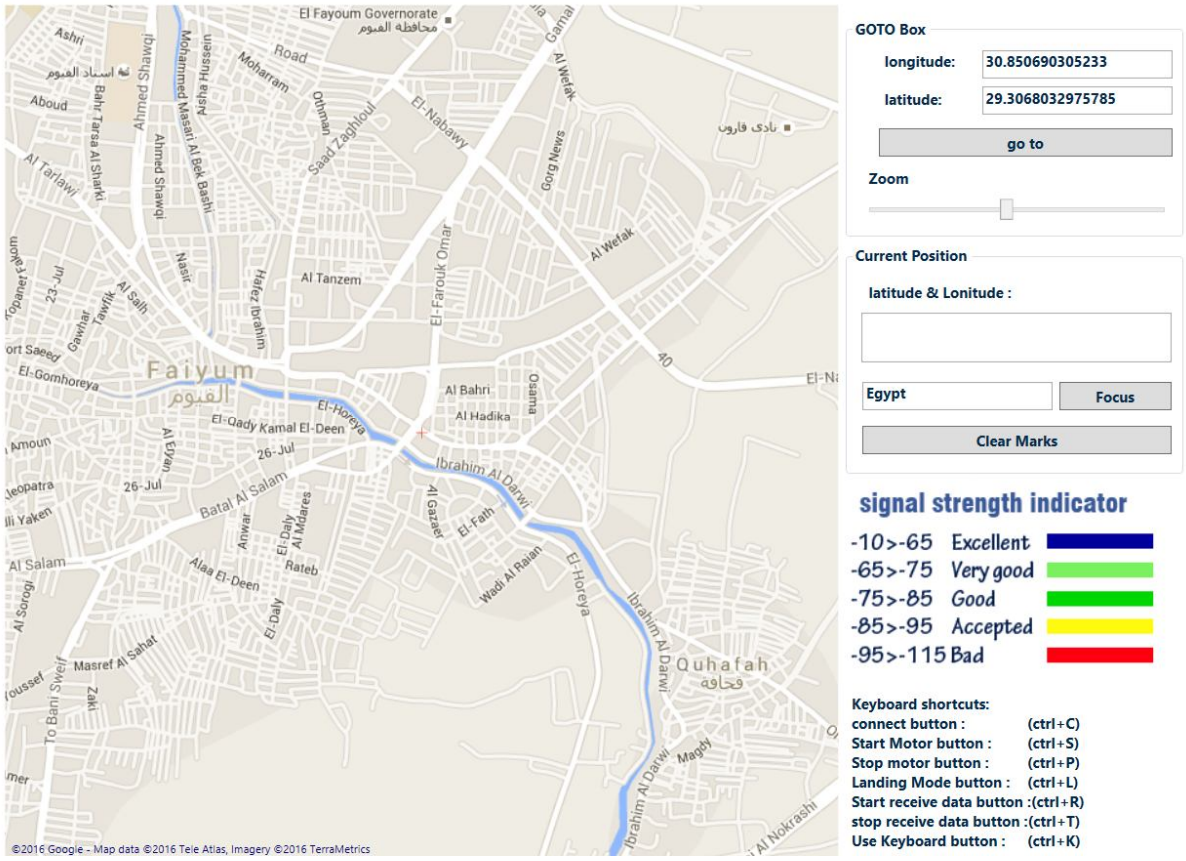


Figure ٣٩ area ٤ for location data

Object	Object Name	Action	notes
Buttons			
go to	btnGoto	Point the position of Lat. & long. On the map	
Clear Marks	btnClearMark	Clear all marks from the map	
slider			
Zoom	sliderZoom	Change the zoom of map	
Text block			
Longitude	txtLongitude	Contain the longitude from GPS	
Latitude	txtLatitude	Contain the Latitude from GPS	
Focus area	txtFoucsArea	Enter the name of area where you want to search on map	
Grid			
map	mapControl	Map provider	

٨.٥ Area num. ٥ : Data Monitoring

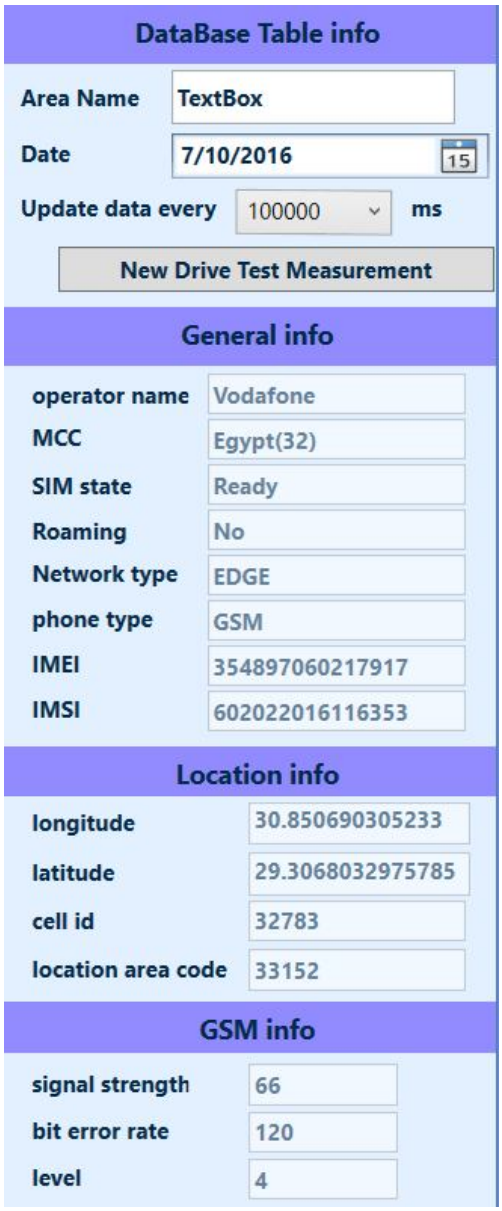


Figure ٤٠ area ٥ for Drive test data monitoring

Object	Object Name	Action	notes

Buttons			
New Drive Test Measurement	btnNewTable	Start data base table	
slider			
Zoom	sliderZoom	Change the zoom of map	
Text block			
Area name	txtAreaName	Enter the area name of measurement	
Date	txtPeriod	Enter Date of the day	(Area name + date) are the name of table
Operator name	txtOperatorName	Show the operator company name	
MCC	txtMcc		
SIM State	txtSimState	Show the state of SIM (ready, block)	
Roaming	txtRoaming	Show the roaming state	
Network type	txtNetworkType	Show the network state (EDGE…..)	
Phone type	txtPhoneType	Show the voice call state (GSM…..)	
IMEI	txtImei	International mobile equipment identity	
IMSI	txtImsi	International Mobile Subscriber Identity	
Dell ID	txtCellId	Show operator cell id	
Location area code	txtLocationAreaCode	Show location cell code	
Signal strength	txtSignalStrength	Show the value of signal strength	
Level	txtLevel	Show the level of signal strength	

٨.٦ Area num. ٥: joystick area

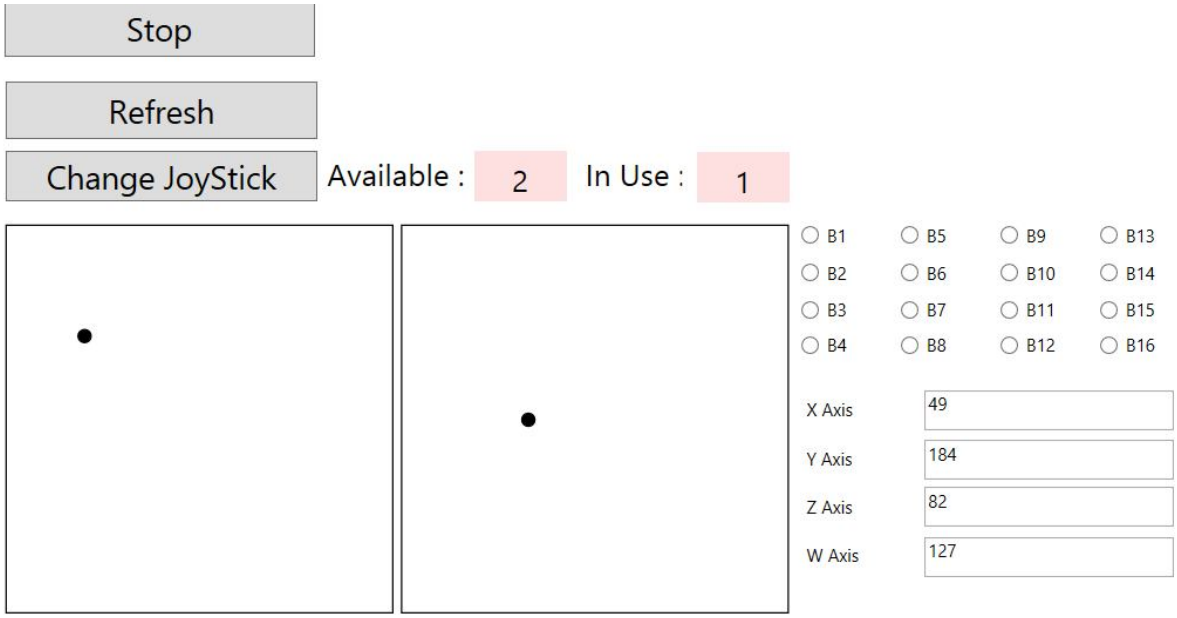


Figure ٤١ area٥ for joystick controlling

Object	Object Name	Action	notes
Buttons			
Start JSK	btnStartJSK	Start the response to the joystick	
JSK refresh	btnJSKrefresh	Refresh the com port	
Change joystick	btnchangejoystick	Choose between different joystick	
Label			
available	lblavailable	Show the number of available joystick	
inuse	lblinuse	Show the number of selected joystick	
Text block			
X axis	tx	Show the motion of left stick(right/left)	
Y axis	ty	Show the motion of left stick(up/down)	
Z axis	tz	Show the motion of right stick(right/left)	
W axis	tw	Show the motion of right stick(up/down)	
Grid			
Right stick	GridZW	Show the motion of right stick	
Left stick	GridXY	Show the motion of Left stick	

٨.٧ Serial Port Command

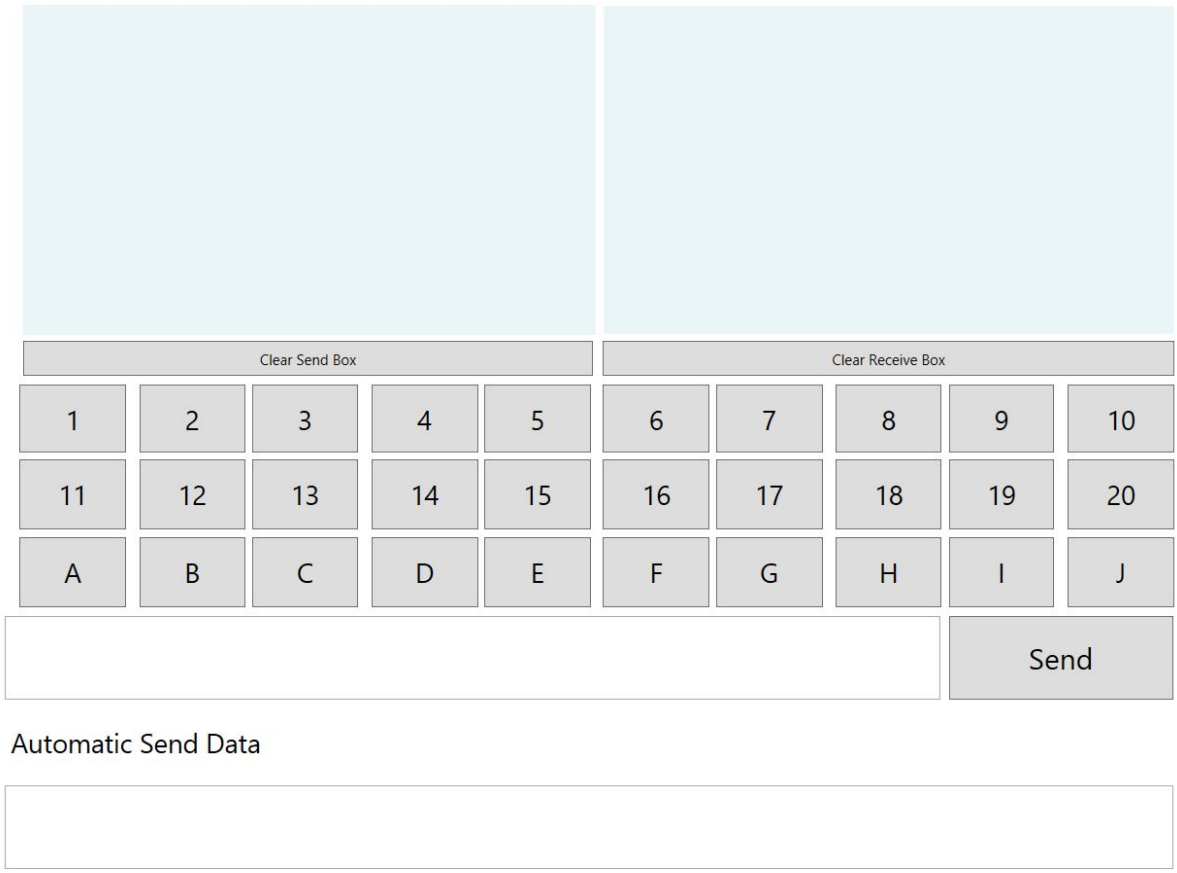


Figure ٤٢ this is for direct Serial communication

Object	Object Name	Action	notes
Buttons			
Send	btnSend	Send data in textbox	
Clear Send Box	clearTX	Clear Send Box	
Clear Receive Box	clearRX	Clear Receive Box	
Text block			
Transmitted data	CommdataTX	Show the Transmitted data	
received data	CommdataRX	Show the received data	
Text Box			
Data to sent	SerialData	Sent on click	
Automatic Send Data	SerialDataAuto	Automatic Send Data in the text box	

٨.٨Data Base window

From it you can see any previous measurement table and see it on map.

It provide also the ability to export data in excel sheet.

Object	Object Name	Action	notes
Buttons			
Refresh DataBase	btnRefreshDataBase	Refresh Data Base table and data	
Delete	btnDeleteTable	Delete selected Table	
Export To Excel	btnExportToExcel	Export selected table to Excel	
View Map	btnViewMap	View Map of this area	
List box			
Transmitted data	lstTableName	Show the table name in Data base	
Data Grid			
Data Grid	DataGrid	Show the data in selected Table	
Grid			
Map Control	mapControl	View the map	

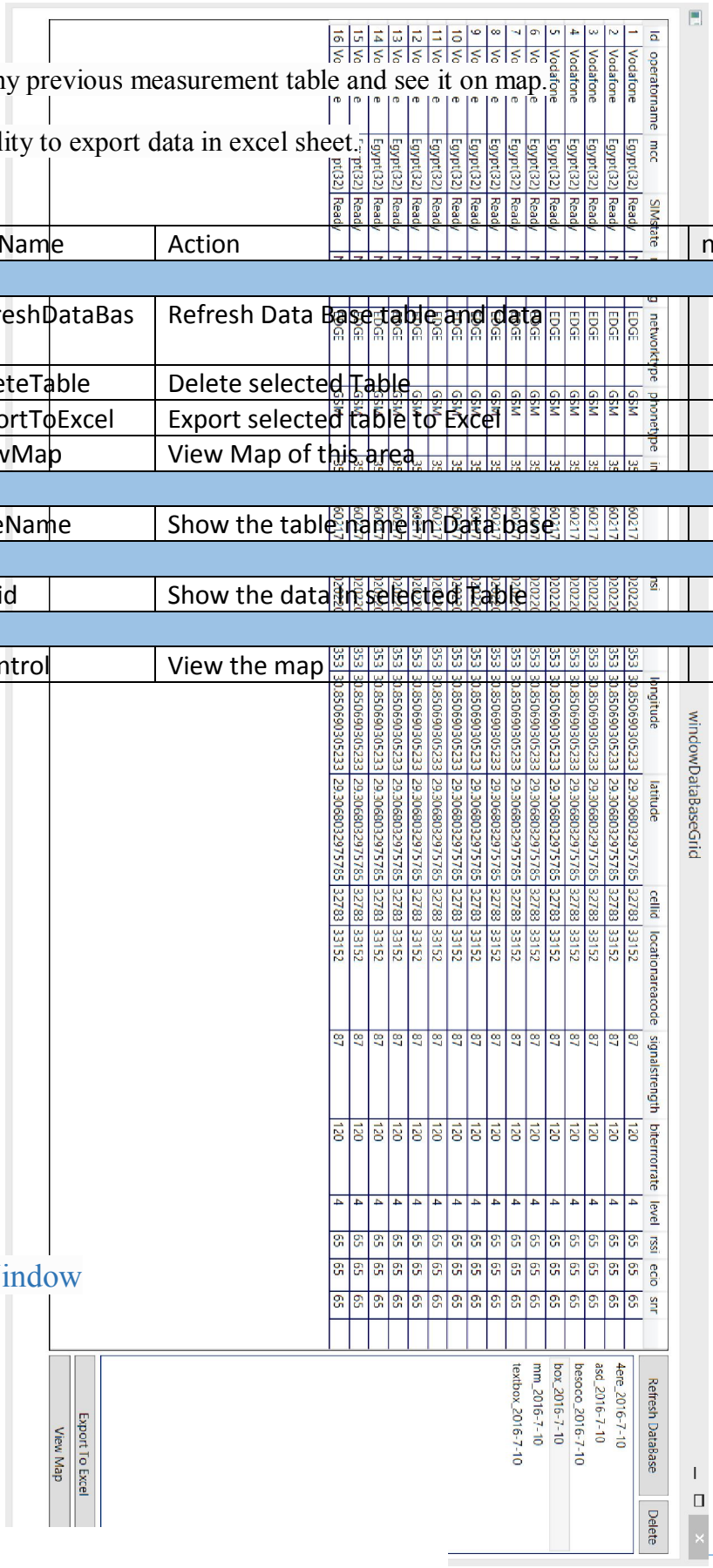


Figure ٤٣ database window

٨.٩Help Window

Show the brief info about app

Figure ٤٤ GUI help window

Chapter Nine

٩. HOW TO USE THE PROJECT

٩.١ Setup

١. Choose serial portin GUI.

٢. Make sure of baud rate of RF, default: ٩٦٠٠ bps.

٣. Click on connect button in GUI.

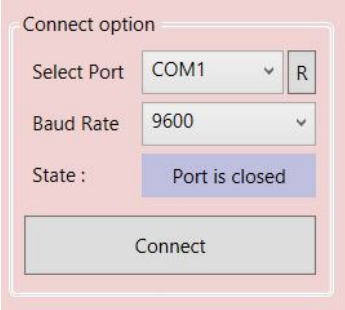


Figure ٤٥ step ٣ in setup

Figure ٤٦ step ٤ in setup

٤. Turn on the remote control.

٥. Choose the required controller upon channel ٥ (switch D).

a. Up for remote control.

b. Down for computer

Figure ٤٧ step ٥ in setup

٦. Turn on arduino Nano of PID calculation.

٧. Make sure that the remote receiver's led is turned on.

٨. Turn on arduino mega.

٩. Open android app.

١٠. Make sure the app is working correctly.

١١. Finally, connect the battery to ESC.

١٢. Move away from the copter.

٩.٢ Start motors

A. From GUI

١. Press the start motors button (ctrl+s).

٢. Press the use keyboard button (ctrl+k).

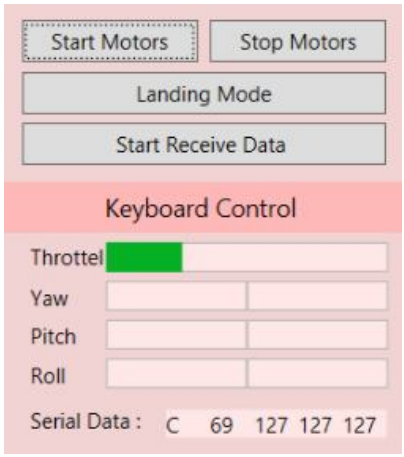


Figure ٤٨ controlling the copter through GUI

٣. Increase the throttle by the sum button on the calculator side of the keyboard.

B. From remote

١. Switch channel ٥ (switch D) up.

٢. Start motor by

a. Put the throttle at minimum value.

b. Then, put the yaw to the minimum value to the left.

Figure ٤٩ starting copter from remote

c. Put the yaw back to initial position.

٩.٣ Flight control

١. Move up

By increasing the throttle channel

GUI	Remote
Press the '+' button on the calculator side of the keyboard.	Increase throttle (channel ٣), the left stick up.

٢. Move down

By decreasing the throttle channel

GUI	Remote
Press the '-' button on the calculator side of the keyboard.	Decrease throttle (channel ٣), the left stick down.

٣. Move front

By increasing the pitch channel

GUI	Remote
Press the '٨' button on the calculator side of the keyboard.	Increase pitch (channel ٢), the right stick up.

٤. Move backward

By decreasing the pitch channel

GUI	Remote
Press the '٢' button on the calculator side of the keyboard.	Decrease pitch (channel ٢), the right stick down.

٥. Move right

By increasing the roll channel

GUI	Remote
Press the '٦' button on the calculator side of the keyboard.	Increase roll (channel ١), the right stick right.

٦. Move left

By decreasing the roll channel

GUI	Remote
Press the '٤' button on the calculator side of the keyboard.	Decrease roll (channel ١), the right stick left.

٧. Rotate clockwise

By increasing the yaw channel

GUI	Remote
Press the '٩' button on the calculator side of the keyboard.	Increase yaw (channel ٤), the left stick right.

٨. Rotate counter-clockwise

By decreasing the yaw channel

GUI	Remote
Press the '٧' button on the calculator side of the keyboard.	Decrease yaw (channel ٤), the left stick left.

٩.٤ Receive data

To receive data press the "start receive" button on the GUI

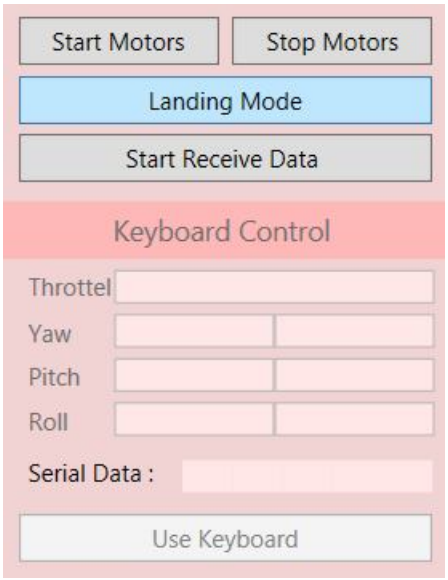


Figure ٥٠ start receive data

Examples of the data received from the copter are copter batteries status, GPS data and mobile drive test data.

٩.٥ Setup the data base table

By entering the area name, date and refresh rate then press the "New drive test

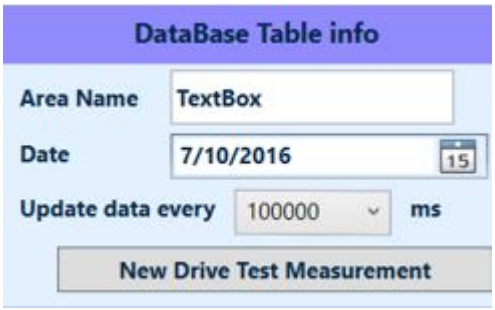


Figure ٥١ start storing in the database

measurement" button.

Now, you are able to fight the copter by remote or computer and you can see the data LIVE updated from the position of the copter on the map and data base viewer

٩.٦ Finishing the mission

١. Stop measurement

٢. Stop receive data

٣. Landing

a. Manually

GUI	Remote
Decrease the throttle until the copter reaches the ground then press "stop motors" button.	Decrease the throttle until the copter reaches the ground then stop the copter by increasing the yaw to the maximum.

b. Automatically

GUI	Remote
Just click the "landing Mode" button.	Increase channel ٦ to its maximum (switch C).

Now, you can have the data from (file -> database). You can see the data and can export it into Excel sheet and see the map location of each row of the data by pressing the "view map" button.

٩.٧Caution and Warnings

١. Safety comes FIRST.

٢. Do not come any close to the copter during flying.

٣. Keep noticing the batteries level on the GUI.

٤. Always start copter outdoor, never work with it indoors.

٥. Do not make transitions of copter controller between Remote and Computer during flying. Always choose your controller while the copter is in the ground.

٦. Always make sure that every connection is properly connected.

Appendix A

GUI Code

Main window

```
١.  using System;
٢.  using System.Collections.Generic;
٣.  using System.Linq;
٤.  using System.Text;
٥.  using System.Threading.Tasks;
٦.  using System.Windows;
٧.  using System.Windows.Controls;
٨.  using System.Windows.Data;
٩.  using System.Windows.Documents;
١٠. using System.Windows.Input;
١١. using System.Windows.Media;
١٢. using System.Windows.Media.Imaging;
١٣. using System.Windows.Navigation;
١٤. using System.Windows.Shapes;
١٥. using System.IO.Ports;
١٦. using System.Threading;
١٧. using System.Windows.Threading;
١٨. using SlimDX.DirectInput;
١٩. using System.Timers;
٢٠. using GMap.NET.MapProviders;
٢١. using GMap.NET;
٢٢. using GMap.NET.WindowsPresentation;
٢٣. using Microsoft.Win32;
٢٤.
٢٥.
٢٦. namespace SerialPortControl
٢٧. {
٢٨.
٢٩.
٣٠.
٣١.     public partial class MainWindow : Window
٣٢.     {
٣٣.
٣٤.
٣٥.         #region variables
٣٦.         //Richtextbox
٣٧.         FlowDocument mcFlowDoc = new FlowDocument();
٣٨.         Paragraph para = new Paragraph();
٣٩.         FlowDocument mcFlowDoc2 = new FlowDocument();
٤٠.         Paragraph para2 = new Paragraph();
٤١.
٤٢.         //Serial
```

```
٤٣.         SerialPort serial = new SerialPort();
٤٤.         string recieved_data;
٤٥.
٤٦.         bool connect_state = false;
٤٧.         bool JSK_state = false;
٤٨.
٤٩.         int jskInUse = 0;
٥٠.
٥١.         int throttel = 1, yaw = 127, pitch = 127, roll = 127;
٥٢.         char firstchar = 'C';
٥٣.
٥٤.         bool keyboard_selected = false;
٥٥.
٥٦.         //recieve indicator
٥٧.         bool get_G = false, get_D = false, get_H = false, get_B
      = false, get_E = false, get_M = false, get_N = false, get_A =
      false, get_F = false, get_W = false, get_I = false;
٥٨.         bool get_a = false, get_b = false, get_c = false, get_d
      = false, get_o = false, get_f = false, get_g = false, get_i =
      false, get_h = false, get_j = false, get_k = false, get_l = fa
      lse, get_m = false, get_n = false;
٥٩.         #endregion
٦٠.
٦١.
٦٢.
٦٣.         #region Recieving
٦٤.
٦٥.         private delegate void UpdateUiTextDelegate(string text);

٦٦.         private void Recieve(object sender, System.IO.Ports.Seri
      alDataReceivedEventArgs e)
٦٧.         {
٦٨.             try
٦٩.             {
٧٠.                 if (serial.IsOpen)
٧١.                 {
٧٢.                     char RX = (char)serial.ReadChar();
٧٣.                     map_recieved_data(RX);
٧٤.                     // Collecting the characters received to our
       'buffer' (string).
٧٥.                     //recieved_data = serial.ReadExisting();
٧٦.                     // Dispatcher.Invoke(DispatcherPriority.Send
      , new UpdateUiTextDelegate(WriteData), recieved_data.ToString(
      ));
٧٧.                 }
٧٨.
٧٩.             }
٨٠.             catch (TimeoutException ex) { MessageBox.Show(ex.ToS
      tring()); }
```

```
٨١.         }
٨٢.         private void WriteData(string text)
٨٣.         {
٨٤.             try
٨٥.             {
٨٦.                 // Assign the value of the recieved_data to the
      RichTextBox.
٨٧.                 para.Inlines.Add(text);
٨٨.                 mcFlowDoc.Blocks.Add(para);
٨٩.                 CommdataRX.Document = mcFlowDoc;
٩٠.             }
٩١.             catch (TimeoutException ex)
٩٢.             {
٩٣.                 MessageBox.Show(ex.Message);
٩٤.             }
٩٥.
٩٦.
٩٧.         }
٩٨.
٩٩.         #endregion
١٠٠.
١٠١.           #region Sending
١٠٢.
١٠٣.
١٠٤.           private void Send_Data(object sender, RoutedEventArgs
      e)
١٠٥.           {
١٠٦.               SerialCmdSend(SerialData.Text);
١٠٧.               SerialData.Text = "";
١٠٨.           }
١٠٩.           public void SerialCmdSend(string data)
١١٠.           {
١١١.               if (serial.IsOpen)
١١٢.               {
١١٣.                   try
١١٤.                   {
١١٥.
١١٦.                       // Send the binary data out the port
١١٧.                       byte[] hexstring = Encoding.ASCII.GetBytes
      (data);
١١٨.                       //There is a intermitant problem that I ca
      me across
١١٩.                       //If I write more than one byte in succesi
      on without a
١٢٠.                       //delay the PIC i'm communicating with wil
      l Crash
١٢١.                       //I expect this id due to PC timing issues
       ad they are
```

```
                        //not directley connected to the COM port the solution
                        //Is a ver small 1 millisecound delay between chracters
                        foreach (byte hexval in hexstring)
                        {
                            byte[] _hexval = new byte[] { hexval }; // need to convert byte to byte[] to write
                            serial.Write(_hexval, 0, 1);
                            Thread.Sleep(1);
                        }
                        CommdataTX.Text += data;
                    }
                    catch (TimeoutException ex)
                    {
                        MessageBox.Show(ex.Message);
                    }
                }
                else
                {
                }
            }

            #endregion

            #region Joystick Function

            private int _x;
            private int _y;
            DirectInput input = new DirectInput();
            SlimDX.DirectInput.Joystick stick;
            Joystick[] sticks;
            int yvalue = 0;
            int xvalue = 0;
            int zvalue = 0;
            int wvalue = 0;
            public Joystick[] getsticks()
            {
                List<SlimDX.DirectInput.Joystick> sticks = new List<SlimDX.DirectInput.Joystick>();
                foreach (DeviceInstance device in input.GetDevices(DeviceClass.GameController, DeviceEnumerationFlags.AttachedOnly))
                {
                    try
                    {
                        stick = new SlimDX.DirectInput.Joystick(input, device.InstanceGuid);
                        stick.Acquire();
```

```
                    foreach (DeviceObjectInstance deviceobject in stick.GetObjects())
                    {
                        if ((deviceobject.ObjectType & ObjectDeviceType.Axis) != 0)
                        {
                            stick.GetObjectPropertiesById((int)deviceobject.ObjectType).SetRange(0, 255);
                        }

                    }
                    sticks.Add(stick);

                }
                catch (DirectInputException)
                {

                }
            }
            return sticks.ToArray();

        }

        void stickHandle(Joystick stick, int id)
        {

            if (id == jskInUse)
            {
                JoystickState state = new JoystickState();
                state = stick.GetCurrentState();
                xvalue = state.X;
                yvalue = 255 - state.Y;
                wvalue = 255 - state.Z;
                zvalue = state.RotationZ;

                tx.Text = xvalue.ToString();
                ty.Text = yvalue.ToString();
                tz.Text = zvalue.ToString();
                tw.Text = wvalue.ToString();

                rec1.Margin = new Thickness((xvalue), (255 - yvalue), 0, 0);
                rec2.Margin = new Thickness(zvalue, (255 - wvalue), 0, 0);

                bool[] buttons = state.GetButtons();

                if (buttons[0]) { b1.IsChecked = true; }
```

```
                    if (buttons[1]) { b2.IsChecked = true; }
                    if (buttons[2]) { b3.IsChecked = true; }
                    if (buttons[3]) { b4.IsChecked = true; }
                    if (buttons[4]) { b5.IsChecked = true; }
                    if (buttons[5]) { b6.IsChecked = true; }
                    if (buttons[6]) { b7.IsChecked = true; }
                    if (buttons[7]) { b8.IsChecked = true; }
                    if (buttons[8]) { b9.IsChecked = true; }
                    if (buttons[9]) { b10.IsChecked = true; }
                    if (buttons[10]) { b11.IsChecked = true; }
                    if (buttons[11]) { b12.IsChecked = true; }
                    if (buttons[12]) { b13.IsChecked = true; }
                    if (buttons[13]) { b14.IsChecked = true; }
                    if (buttons[14]) { b15.IsChecked = true; }
                    if (buttons[15]) { b16.IsChecked = true; }

                }

                send_jsk_parameter(firstchar, xvalue, yvalue, zval
ue, wvalue);
            }
            #endregion

            #region Timer
            private void dispatcherTimer_Tick(object sender, Event
Args e)
            {
                if (JSK_state == true)
                {
                    for (int i = 0; i < sticks.Length; i++)
                    {
                        stickHandle(sticks[i], i);
                    }
                    if (sticks.Length > 0)
                    {

                        if (jskInUse > (sticks.Length - 1))
                        {
                            jskInUse = 0;
                        }
                        lblinuse.Content = (jskInUse + 1).ToString
();
                    }
                    else { lblinuse.Content = "No"; }

                    _x = xvalue;
                    _y = yvalue;
                }
```

```
            //else if (keyboard_selected == true && connect_st
ate == true)
            //{
            //    send_jsk_parameter(firstchar, throttel, yaw,
 pitch, roll);

            //}

            if (connect_state == true && serial.IsOpen == fals
e)
            {
                btnConnect_Click(null, null);
                reset_keyboard();
                MessageBox.Show("Connection Has been lost");

            }


        }

        // timer to update data
        private void DataUpdateTimer_Tick(object sender, Event
Args e)
        {
            if (keyboard_selected == true && connect_state ==
true)
            {
                send_jsk_parameter(firstchar, throttel, yaw, p
itch, roll);

            }

        }
        #endregion

        #region Function
        void send_jsk_parameter(Char firstchar, int throttle,
int yaw, int pitch, int roll)
        {
            if (serial.IsOpen)
            {
                /*  byte channel_1 = (byte)firstchar;
                  byte channel_2 = (byte)throttle;
                  byte channel_3 = (byte)yaw;
                  byte channel_4 = (byte)pitch;
                  byte channel_5 = (byte)roll;
                  byte[] SerialData = new byte[5] { channel_1,
 channel_5, channel_4, channel_2, channel_3 };
                  serial.Write(SerialData, 0, 5);*/
```

```
                    string char_roll = "r"; string char_pitch = "p
    "; string char_throttle = "t"; string char_yaw = "y";

                    string channel_1 = IntToString(roll);
                    string channel_2 = IntToString(pitch);
                    string channel_3 = IntToString(throttle);
                    string channel_4 = IntToString(yaw);
                    SerialCmdSend(char_roll + channel_1 + char_pit
    ch + channel_2 + char_throttle + channel_3 + char_yaw + channe
    l_4 + firstchar.ToString());
                }
            }

            public static string IntToString(int intNum)
            {
                string DataInString = "";
                if (intNum >= 0 && intNum <= 9)
                {
                    DataInString = "00" + intNum.ToString();
                }
                else if (intNum > 9 && intNum <= 99)
                {
                    DataInString = "0" + intNum.ToString();
                }
                else if (intNum > 99 && intNum <= 225)
                {
                    DataInString = intNum.ToString();
                }
                return DataInString;
            }

            void get_Comport_name()
            {
                CboxCom.Items.Clear();
                string[] ports = SerialPort.GetPortNames();
                foreach (string port in ports)
                {
                    CboxCom.Items.Add(port);
                }

            }
            void get_baud_rate()
            {
                CboxBaudRate.Items.Clear();
                string[] baud = { "1200", "2400", "4800", "9600",
    "19200", "38400", "57600", "74800", "115200", "230400", "25000
    0" };
                foreach (string Rate in baud)
                {
```

```
٣٣٨.                    CboxBaudRate.Items.Add(Rate);
٣٣٩.                }
٣٤٠.            }
٣٤١.
٣٤٢.            int counter = 0;
٣٤٣.            char[] inputdata = new char[20];
٣٤٤.            void map_recieved_data(char Data)
٣٤٥.            {
٣٤٦.                switch (Data)
٣٤٧.                {
٣٤٨.                    case 'G': get_variable_method(); get_G = true;
        break;
٣٤٩.                    case 'D': get_variable_method(); get_D = true;
        break;
٣٥٠.                    case 'F': get_variable_method(); get_F = true;
        break;
٣٥١.                    case 'H': get_variable_method(); get_H = true;
        break;
٣٥٢.                    case 'B': get_variable_method(); get_B = true;
        break;
٣٥٣.                    case 'E': get_variable_method(); get_E = true;
        break;
٣٥٤.                    case 'M': get_variable_method(); get_M = true;
        break;
٣٥٥.                    case 'N': get_variable_method(); get_N = true;
        break;
٣٥٦.                    case 'A': get_variable_method(); get_A = true;
        break;
٣٥٧.                    case 'W': get_variable_method(); get_W = true;
        break;
٣٥٨.                    case 'a': get_variable_method(); get_a = true;
        break;
٣٥٩.                    case 'b': get_variable_method(); get_b = true;
        break;
٣٦٠.                    case 'c': get_variable_method(); get_c = true;
        break;
٣٦١.                    case 'd': get_variable_method(); get_d = true;
        break;
٣٦٢.                    case 'o': get_variable_method(); get_o = true;
        break;
٣٦٣.                    case 'f': get_variable_method(); get_f = true;
        break;
٣٦٤.                    case 'g': get_variable_method(); get_g = true;
        break;
٣٦٥.                    case 'i': get_variable_method(); get_i = true;
        break;
٣٦٦.                    case 'h': get_variable_method(); get_h = true;
        break;
```

```
٣٦٧.                    case 'j': get_variable_method(); get_j = true;
    break;
٣٦٨.                    case 'k': get_variable_method(); get_k = true;
    break;
٣٦٩.                    case 'l': get_variable_method(); get_l = true;
    break;
٣٧٠.                    case 'm': get_variable_method(); get_m = true;
    break;
٣٧١.                    case 'n': get_variable_method(); get_n = true;
    break;
٣٧٢.
٣٧٣.                    default:
٣٧٤.                        if ((Data >= 48) && (Data <= 57) || Data =
    = 46)
٣٧٥.                        {
٣٧٦.                            if (get_G) { inputdata[counter] = Data
    ; counter++; }
٣٧٧.                            else if (get_D) { inputdata[counter] =
     Data; counter++; }
٣٧٨.                            else if (get_F) { inputdata[counter] =
     Data; counter++; }
٣٧٩.                            else if (get_H) { inputdata[counter] =
     Data; counter++; }
٣٨٠.                            else if (get_B) { inputdata[counter] =
     Data; counter++; }
٣٨١.                            else if (get_E) { inputdata[counter] =
     Data; counter++; }
٣٨٢.                            else if (get_M) { inputdata[counter] =
     Data; counter++; }
٣٨٣.                            else if (get_N) { inputdata[counter] =
     Data; counter++; }
٣٨٤.                            else if (get_A) { inputdata[counter] =
     Data; counter++; }
٣٨٥.                            else if (get_W) { inputdata[counter] =
     Data; counter++; }
٣٨٦.                            else if (get_a) { inputdata[counter] =
     Data; counter++; }
٣٨٧.                            else if (get_b) { inputdata[counter] =
     Data; counter++; }
٣٨٨.                            else if (get_c) { inputdata[counter] =
     Data; counter++; }
٣٨٩.                            else if (get_d) { inputdata[counter] =
     Data; counter++; }
٣٩٠.                            else if (get_o) { inputdata[counter] =
     Data; counter++; }
٣٩١.                            else if (get_f) { inputdata[counter] =
     Data; counter++; }
٣٩٢.                            else if (get_g) { inputdata[counter] =
     Data; counter++; }
```

```
٣٩٣.                        else if (get_i) { inputdata[counter] =
      Data; counter++; }
٣٩٤.                        else if (get_h) { inputdata[counter] =
      Data; counter++; }
٣٩٥.                        else if (get_j) { inputdata[counter] =
      Data; counter++; }
٣٩٦.                        else if (get_k) { inputdata[counter] =
      Data; counter++; }
٣٩٧.                        else if (get_l) { inputdata[counter] =
      Data; counter++; }
٣٩٨.                        else if (get_m) { inputdata[counter] =
      Data; counter++; }
٣٩٩.                        else if (get_n) { inputdata[counter] =
      Data; counter++; }
٤٠٠.
٤٠١.                    }
٤٠٢.
٤٠٣.                    break;
٤٠٤.
٤٠٥.                }
٤٠٦.            }
٤٠٧.            void write_recieve_data(string data)
٤٠٨.            {
٤٠٩.                if (get_G) { txtLatitude.Text = data; spDT.txtLati
      tude.Text = data; btnGoto_Click(null, null); }
٤١٠.                else if (get_D) { txtLongitude.Text = data; spDT.t
      xtLongitude.Text = data; btnGoto_Click(null, null); }
٤١١.                else if (get_H) { lblQuadBattery.Content = (int.Pa
      rse(data)).ToString() + "%"; pBQuadBattery.Value = int.Parse(d
      ata); }
٤١٢.                else if (get_B) { lblController1Battery.Content =
      (int.Parse(data)).ToString() + "%"; pBController1Battery.Value
       = int.Parse(data); }
٤١٣.                else if (get_F) { lblController2Battery.Content =
      (int.Parse(data)).ToString() + "%"; pBController2Battery.Value
       = int.Parse(data); }
٤١٤.                else if (get_M) { txtSatSpeed.Text = (int.Parse(da
      ta)).ToString(); }
٤١٥.                else if (get_N) { txtSatNum.Text = (int.Parse(data
      )).ToString(); }
٤١٦.                else if (get_E) { txtError.Text = txtError.Text +
      data; }
٤١٧.                else if (get_A) { txtSatAltitude.Text = (int.Parse
      (data)).ToString(); }
٤١٨.                else if (get_W) { txtulterasonic.Text = (int.Parse
      (data)).ToString(); }
٤١٩.                else if (get_a) { spDT.txtSignalStrength.Text = "-
      " + data; }
٤٢٠.                else if (get_b) { spDT.txtLevel.Text = data; }
```

```
٤٢١.                else if (get_c) { spDT.txtCellId.Text = data; }
٤٢٢.                else if (get_d) { spDT.txtLocationAreaCode.Text =
      data; }
٤٢٣.                else if (get_o) { spDT.txtOperatorName.Text = data
      ; }
٤٢٤.                else if (get_f) { spDT.txtMcc.Text = data; }
٤٢٥.                else if (get_g) { spDT.txtSimState.Text = data; }

٤٢٦.                else if (get_i) { spDT.txtRoaming.Text = get_Roami
      ng(data); }
٤٢٧.                else if (get_h) { spDT.txtNetworkType.Text = get_N
      etwork_type(data); }
٤٢٨.                else if (get_j) { spDT.txtPhoneType.Text = get_pho
      ne_type(data); }
٤٢٩.                else if (get_k) { spDT.txtImei.Text = data; }
٤٣٠.                else if (get_l) { spDT.txtImsi.Text = data; }
٤٣١.                else if (get_m) { spDT.txtvoiceCapable.Text = get_
      voice_capable(data); }
٤٣٢.                else if (get_n) {spDT.txtSimSerial.Text=data; }
٤٣٣.            }
٤٣٤.            void reset_keyboard()
٤٣٥.            {
٤٣٦.                throttel = 0; yaw = 127; pitch = 127; roll = 127;

٤٣٧.                firstchar = 'C';
٤٣٨.                pBThrottel.Value = throttel;
٤٣٩.            }
٤٤٠.            void get_variable_method()
٤٤١.            {
٤٤٢.                char[] buffer = new char[counter];
٤٤٣.                buffer = inputdata;
٤٤٤.                string str = new string(buffer);
٤٤٥.                str = str.Replace("\0", string.Empty);
٤٤٦.                Dispatcher.Invoke(DispatcherPriority.Send, new Upd
      ateUiTextDelegate(write_recieve_data), str.ToString());
٤٤٧.                get_G = false; get_D = false; get_H = false; get_B
       = false; get_E = false; get_M = false; get_N = false; get_A =
       false; get_F = false; get_W = false; get_I = false;
٤٤٨.                get_a = false; get_b = false; get_c = false; get_d
       = false; get_o = false; get_f = false; get_g = false; get_i =
       false; get_h = false; get_j = false; get_k = false; get_l = f
      alse; get_m = false; get_n = false;
٤٤٩.                counter = 0;
٤٥٠.                inputdata = new char[20];
٤٥١.            }
٤٥٢.            string get_Network_type(string indicator)
٤٥٣.            {
٤٥٤.                int NET=int.Parse(indicator);
٤٥٥.                string networkType = "UNKNOWN";
```

```
                    if (NET == 0){networkType="UNKNOWN";}
                    else if (NET == 1) { networkType = "GPRS"; }
                    else if (NET == 2) { networkType = "EDGE"; }
                    else if (NET == 3) { networkType = "UMTS"; }
                    else if (NET == 4) { networkType = "CDMA"; }
                    else if (NET == 5) { networkType = "EVDO revis
    ion 0"; }
                    else if (NET == 6) { networkType = "EVDO revis
    ion A"; }
                    else if (NET == 7) { networkType = "1xRTT"; }

                    else if (NET == 6) { networkType = "HSDPA"; }

                    else if (NET == 9) { networkType = "HSUPA"; }

                    else if (NET == 10) { networkType = "HSPA"; }

                    else if (NET == 11) { networkType = "IDEN"; }

                    else if (NET == 12) { networkType = "EVDO revi
    sion B"; }
                    else if (NET == 13) { networkType = "LTE "; }

                    else if (NET == 14) { networkType = "EHRPD"; }

                    else if (NET == 15) { networkType = "HSPAP"; }

                    else                { networkType = "UNKNOWN";
     }
                return networkType;
            }
            string get_phone_type(string indicator)
            {
                int NET = int.Parse(indicator);
                string phoneType = "NONE";
                if (NET == 0) { phoneType = "NONE"; }
                else if (NET == 1) { phoneType = "GSM"; }
                else if (NET == 2) { phoneType = "CDMA"; }
                else if (NET == 3) { phoneType = "SIP"; }

                return phoneType;
            }
            string get_voice_capable(string indicator)
            {
                int NET = int.Parse(indicator);
                string phoneType = "NONE";
                if (NET == 0) { phoneType = "NONE"; }
                else if (NET == 1) { phoneType = "Data Only"; }
                return phoneType;
```

```
٤٩٣.          }
٤٩٤.          string get_Roaming(string indicator)
٤٩٥.          {
٤٩٦.              int NET = int.Parse(indicator);
٤٩٧.              string phoneType = "NONE";
٤٩٨.              if (NET == 0) { phoneType = "Roaming"; }
٤٩٩.              else if (NET == 1) { phoneType = "No Roaming"; }
٥٠٠.              return phoneType;
٥٠١.          }
٥٠٢.          #endregion
٥٠٣.
٥٠٤.          controllerDriveTest spDT = new controllerDriveTest();

٥٠٥.
٥٠٦.          public MainWindow()
٥٠٧.          {
٥٠٨.              InitializeComponent();
٥٠٩.              get_Comport_name();
٥١٠.              get_baud_rate();
٥١١.              GridSendReceive.IsEnabled = false;
٥١٢.              GridBtn.IsEnabled = false;
٥١٣.
٥١٤.              #region jotstickParameter
٥١٥.
٥١٦.              System.Windows.Threading.DispatcherTimer dispatche
      rTimer = new System.Windows.Threading.DispatcherTimer();
٥١٧.              dispatcherTimer.Tick += dispatcherTimer_Tick;
٥١٨.              dispatcherTimer.Interval = new TimeSpan(0, 0, 0, 0
      , 1);
٥١٩.              dispatcherTimer.Start();
٥٢٠.
٥٢١.
٥٢٢.              System.Windows.Threading.DispatcherTimer DataUpdat
      eTimer = new System.Windows.Threading.DispatcherTimer();
٥٢٣.              DataUpdateTimer.Tick += DataUpdateTimer_Tick;
٥٢٤.              DataUpdateTimer.Interval = new TimeSpan(0, 0, 0, 0
      , 500);
٥٢٥.              DataUpdateTimer.Start();
٥٢٦.
٥٢٧.              _x = 50;
٥٢٨.              _y = 50;
٥٢٩.              getsticks();
٥٣٠.              sticks = getsticks();
٥٣١.              lblavailable.Content = sticks.Length.ToString();
٥٣٢.
٥٣٣.              #endregion
٥٣٤.
٥٣٥.              spDriveTest.Children.Add(spDT);
٥٣٦.
```

```
٥٣٧.             mapControl.MapProvider = GoogleMapProvider.Instanc
     e;
٥٣٨.             mapControl.Position = new PointLatLng(29.306803297
     5785, 30.850690305233);
٥٣٩.             mapControl.Zoom = 15;
٥٤٠.             mapControl.DragButton = MouseButton.Right;
٥٤١.         }
٥٤٢.
٥٤٣.
٥٤٤.         private void mapControl_MouseLeftButtonDown(object sen
     der, MouseButtonEventArgs e)
٥٤٥.         {
٥٤٦.             PointLatLng selectpoint = mapControl.FromLocalToLa
     tLng((int)e.GetPosition(mapControl).X, (int)e.GetPosition(mapC
     ontrol).Y);
٥٤٧.             txtLongLatit.Text = selectpoint.ToString();
٥٤٨.
٥٤٩.             //  txtLatitude.Text = selectpoint.Lat.ToString();

٥٥٠.             //txtLongitude.Text = selectpoint.Lng.ToString();

٥٥١.
٥٥٢.
٥٥٣.
٥٥٤.
٥٥٥.             /*    GMapMarker marker = new GMapMarker(selectpoi
     nt);
٥٥٦.                 Ellipse temp = new Ellipse();
٥٥٧.                 temp.Height = 5;
٥٥٨.                 temp.Width = 5;
٥٥٩.                 temp.Fill = Brushes.Black;
٥٦٠.
٥٦١.                 Image i = new Image();
٥٦٢.                 i.Width = 20;
٥٦٣.                 i.Margin = new Thickness(-10);
٥٦٤.                 BitmapImage bi = new BitmapImage();
٥٦٥.                 bi.BeginInit();
٥٦٦.                 bi.UriSource = new Uri("mark.png", UriKind.Rel
     ative);
٥٦٧.                 bi.EndInit();
٥٦٨.                 i.Source = bi;
٥٦٩.
٥٧٠.
٥٧١.                 marker.Shape = i;*/
٥٧٢.             //mapControl.Markers.Clear();
٥٧٣.             // mapControl.Markers.Add(marker);
٥٧٤.
٥٧٥.         }
٥٧٦.
```

```
٥٧٧.          private void Slider_ValueChanged(object sender, Routed
      PropertyChangedEventArgs<double> e)
٥٧٨.          {
٥٧٩.              mapControl.Zoom = sliderZoom.Value;
٥٨٠.          }
٥٨١.
٥٨٢.          private void btnRefresh_Click(object sender, RoutedEve
      ntArgs e)
٥٨٣.          {
٥٨٤.              try
٥٨٥.              {
٥٨٦.                  get_Comport_name();
٥٨٧.                  CboxCom.SelectedIndex = 0;
٥٨٨.              }
٥٨٩.              catch (Exception ex)
٥٩٠.              {
٥٩١.                  MessageBox.Show(ex.Message);
٥٩٢.              }
٥٩٣.          }
٥٩٤.
٥٩٥.          private void btnConnect_Click(object sender, RoutedEve
      ntArgs e)
٥٩٦.          {
٥٩٧.
٥٩٨.              if (connect_state == false)
٥٩٩.              {
٦٠٠.                  try
٦٠١.                  {
٦٠٢.                      //Sets up serial port
٦٠٣.                      serial.PortName = CboxCom.Text;
٦٠٤.                      serial.BaudRate = Convert.ToInt32(CboxBaud
      Rate.Text);
٦٠٥.                      serial.Handshake = System.IO.Ports.Handsha
      ke.None;
٦٠٦.                      serial.Parity = Parity.None;
٦٠٧.                      serial.DataBits = 8;
٦٠٨.                      serial.StopBits = StopBits.One;
٦٠٩.                      serial.ReadTimeout = 2000;
٦١٠.                      serial.WriteTimeout = 500;
٦١١.                      serial.Open();
٦١٢.                      connect_state = true;
٦١٣.                      //Sets button State and Creates function c
      all on data recieved
٦١٤.                      btnConnect.Content = "Disconnect";
٦١٥.                      lblState.Content = "Port is opened";
٦١٦.                      CboxCom.IsEnabled = false;
٦١٧.                      CboxBaudRate.IsEnabled = false;
٦١٨.                      btnRefresh.IsEnabled = false;
٦١٩.                      GridSendReceive.IsEnabled = true;
```

```
                        GridBtn.IsEnabled = true;
                        serial.DataReceived += new System.IO.Ports
.SerialDataReceivedEventHandler(Recieve);
                        reset_keyboard();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show(ex.Message);
                    }
                }
                else if (connect_state == true)
                {
                    try // just in case serial port is not open co
uld also be acheved using if(serial.IsOpen)
                    {
                        serial.Close();
                        connect_state = false;
                        btnConnect.Content = "Connect";
                        lblState.Content = "Port is closed";
                        CboxCom.IsEnabled = true;
                        CboxBaudRate.IsEnabled = true;
                        btnRefresh.IsEnabled = true;
                        GridSendReceive.IsEnabled = false;
                        GridBtn.IsEnabled = false;
                        reset_keyboard();

                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show(ex.Message);
                    }
                }
            }

            private void Button_Click(object sender, RoutedEventAr
gs e)
            {
                if (connect_state == true)
                {

                    string Data = ((Button)sender).Content.ToStrin
g();
                    SerialCmdSend(Data);
                }
            }

            private void btnSend_Click(object sender, RoutedEventA
rgs e)
            {
```

```
٦٦٤.                SerialCmdSend(SerialData.Text);
٦٦٥.                SerialData.Text = "";
٦٦٦.            }
٦٦٧.
٦٦٨.            private void Button_PreviewMouseLeftButtonDown(object
        sender, MouseButtonEventArgs e)
٦٦٩.            {
٦٧٠.                SerialCmdSend("O");
٦٧١.            }
٦٧٢.
٦٧٣.            private void Button_PreviewMouseLeftButtonUp(object se
        nder, MouseButtonEventArgs e)
٦٧٤.            {
٦٧٥.                SerialCmdSend("P");
٦٧٦.            }
٦٧٧.
٦٧٨.            private void btnBuzzer2_PreviewMouseLeftButtonDown(obj
        ect sender, MouseButtonEventArgs e)
٦٧٩.            {
٦٨٠.                SerialCmdSend("Q");
٦٨١.            }
٦٨٢.
٦٨٣.            private void btnBuzzer2_PreviewMouseLeftButtonUp(objec
        t sender, MouseButtonEventArgs e)
٦٨٤.            {
٦٨٥.                SerialCmdSend("R");
٦٨٦.            }
٦٨٧.
٦٨٨.            private void BtnBuzzer3_PreviewMouseLeftButtonDown(obj
        ect sender, MouseButtonEventArgs e)
٦٨٩.            {
٦٩٠.                SerialCmdSend("S");
٦٩١.            }
٦٩٢.
٦٩٣.            private void BtnBuzzer3_PreviewMouseLeftButtonUp(objec
        t sender, MouseButtonEventArgs e)
٦٩٤.            {
٦٩٥.                SerialCmdSend("T");
٦٩٦.            }
٦٩٧.
٦٩٨.            private void clearRX_Click(object sender, RoutedEventA
        rgs e)
٦٩٩.            {
٧٠٠.                CommdataTX.Text = "";
٧٠١.            }
٧٠٢.
٧٠٣.            private void clearTX_Click(object sender, RoutedEventA
        rgs e)
٧٠٤.            {
```

```
٧٠٥.                CommdataRX.Document.Blocks.Clear();
٧٠٦.
٧٠٧.            }
٧٠٨.
٧٠٩.            private void SerialDataAuto_TextChanged(object sender,
         TextChangedEventArgs e)
٧١٠.            {
٧١١.                SerialCmdSend(SerialDataAuto.Text);
٧١٢.                SerialDataAuto.Text = "";
٧١٣.            }
٧١٤.
٧١٥.            private void checkkey_Checked(object sender, RoutedEve
        ntArgs e)
٧١٦.            {
٧١٧.                string Data = ((CheckBox)sender).Content.ToString(
        );
٧١٨.                SerialCmdSend(Data);
٧١٩.            }
٧٢٠.
٧٢١.            private void btnJSKrefresh_Click(object sender, Routed
        EventArgs e)
٧٢٢.            {
٧٢٣.                getsticks();
٧٢٤.                sticks = getsticks();
٧٢٥.                lblavailable.Content = sticks.Length.ToString();
٧٢٦.            }
٧٢٧.
٧٢٨.            private void btnchangejoystick_Click(object sender, Ro
        utedEventArgs e)
٧٢٩.            {
٧٣٠.                if (sticks.Length > 0)
٧٣١.                {
٧٣٢.                    jskInUse++;
٧٣٣.
٧٣٤.                    if (jskInUse > (sticks.Length - 1))
٧٣٥.                    {
٧٣٦.                        jskInUse = 0;
٧٣٧.
٧٣٨.                    }
٧٣٩.                    lblinuse.Content = (jskInUse + 1).ToString();

٧٤٠.                }
٧٤١.                else { lblinuse.Content = "No"; }
٧٤٢.            }
٧٤٣.
٧٤٤.            private void btnStartJSK_Click(object sender, RoutedEv
        entArgs e)
٧٤٥.            {
٧٤٦.                if (JSK_state == false)
```

```
٧٤٧.                {
٧٤٨.                    JSK_state = true;
٧٤٩.                    gridJSKcontrol.IsEnabled = true;
٧٥٠.                    btnStartJSK.Content = "Stop";
٧٥١.                    keyboard_selected = false;
٧٥٢.                    btnKeyboardSelect.Content = "Use Keyboard";
٧٥٣.                    gridKeyboard.IsEnabled = false;
٧٥٤.                }
٧٥٥.                else
٧٥٦.                {
٧٥٧.                    JSK_state = false;
٧٥٨.                    gridJSKcontrol.IsEnabled = false;
٧٥٩.                    btnStartJSK.Content = "Start";
٧٦٠.                    keyboard_selected = true;
٧٦١.
٧٦٢.
٧٦٣.                }
٧٦٤.            }
٧٦٥.
٧٦٦.            private void Window_KeyDown(object sender, KeyEventArg
    s e)
٧٦٧.            {
٧٦٨.                if (e.Key == System.Windows.Input.Key.F1) { Button
    _PreviewMouseLeftButtonDown(null, null); }
٧٦٩.                if (e.Key == System.Windows.Input.Key.F2) { btnBuz
    zer2_PreviewMouseLeftButtonDown(null, null); }
٧٧٠.                if (e.Key == System.Windows.Input.Key.F3) { BtnBuz
    zer3_PreviewMouseLeftButtonDown(null, null); }
٧٧١.                if (e.Key == System.Windows.Input.Key.F5) { btnRef
    resh_Click(null, null); }
٧٧٢.                if (e.Key == System.Windows.Input.Key.T) { btnMoni
    leDataUpdate_Click(null, null); }
٧٧٣.                if (e.Key == System.Windows.Input.Key.C && System.
    Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
    nConnect_Click(null, null); }
٧٧٤.                if (e.Key == System.Windows.Input.Key.S && System.
    Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
    nStartMotor_Click(null, null); }
٧٧٥.                if (e.Key == System.Windows.Input.Key.L && System.
    Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
    nLandingMode_Click(null, null); }
٧٧٦.                if (e.Key == System.Windows.Input.Key.P && System.
    Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
    nStopMotor_Click(null, null); }
٧٧٧.                if (e.Key == System.Windows.Input.Key.K && System.
    Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
    nKeyboardSelect_Click(null, null); }
```

```
٧٧٨.             if (e.Key == System.Windows.Input.Key.R && System.
     Windows.Input.Keyboard.Modifiers == ModifierKeys.Control) { bt
     nStartReceiveData_Click(null, null); }
٧٧٩.             if (keyboard_selected)
٧٨٠.             {
٧٨١.                 if (e.Key == System.Windows.Input.Key.Add) { b
     tnThrottel_PreviewMouseDown(null, null); }
٧٨٢.                 if (e.Key == System.Windows.Input.Key.Subtract
     ) { btnThrotteldown_PreviewMouseDown(null, null); }
٧٨٣.                 if (e.Key == System.Windows.Input.Key.NumPad9)
      { btnYawup_PreviewMouseLeftButtonDown(null, null); }
٧٨٤.                 if (e.Key == System.Windows.Input.Key.NumPad7)
      { btnYawdown_PreviewMouseLeftButtonDown(null, null); }
٧٨٥.                 if (e.Key == System.Windows.Input.Key.NumPad8)
      { btnPitchup_PreviewMouseLeftButtonDown(null, null); }
٧٨٦.                 if (e.Key == System.Windows.Input.Key.NumPad2)
      { btnPitchdown_PreviewMouseLeftButtonDown(null, null); }
٧٨٧.                 if (e.Key == System.Windows.Input.Key.NumPad6)
      { btnRollup_PreviewMouseLeftButtonDown(null, null); }
٧٨٨.                 if (e.Key == System.Windows.Input.Key.NumPad4)
      { btnRolldown_PreviewMouseLeftButtonDown(null, null); }
٧٨٩.             }
٧٩٠.         }
٧٩١.
٧٩٢.         private void Window_KeyUp(object sender, KeyEventArgs
     e)
٧٩٣.         {
٧٩٤.             if (e.Key == System.Windows.Input.Key.F1) { Button
     _PreviewMouseLeftButtonUp(null, null); }
٧٩٥.             if (e.Key == System.Windows.Input.Key.F2) { btnBuz
     zer2_PreviewMouseLeftButtonUp(null, null); }
٧٩٦.             if (e.Key == System.Windows.Input.Key.F3) { BtnBuz
     zer3_PreviewMouseLeftButtonUp(null, null); }
٧٩٧.             if (keyboard_selected)
٧٩٨.             {
٧٩٩.                 if (e.Key == System.Windows.Input.Key.NumPad9)
      { btnYawup_PreviewMouseLeftButtonUp(null, null); }
٨٠٠.                 if (e.Key == System.Windows.Input.Key.NumPad7)
      { btnYawdown_PreviewMouseLeftButtonUp(null, null); }
٨٠١.                 if (e.Key == System.Windows.Input.Key.NumPad8)
      { btnPitchup_PreviewMouseLeftButtonUp(null, null); }
٨٠٢.                 if (e.Key == System.Windows.Input.Key.NumPad2)
      { btnPitchdown_PreviewMouseLeftButtonUp(null, null); }
٨٠٣.                 if (e.Key == System.Windows.Input.Key.NumPad6)
      { btnRollup_PreviewMouseLeftButtonUp(null, null); }
٨٠٤.                 if (e.Key == System.Windows.Input.Key.NumPad4)
      { btnRolldown_PreviewMouseLeftButtonUp(null, null); }
٨٠٥.             }
٨٠٦.         }
```

```
٨٠٧.
٨٠٨.          public void btnThrottel_PreviewMouseDown(object sender
      , MouseButtonEventArgs e)
٨٠٩.          {
٨١٠.              if (throttel >= 0 && throttel < 255)
٨١١.              {
٨١٢.                  throttel++;
٨١٣.              }
٨١٤.              pBThrottel.Value = throttel;
٨١٥.              tbthrottel.Text = throttel.ToString();
٨١٦.              if (keyboard_selected == true && connect_state ==
      true)
٨١٧.              {
٨١٨.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٨١٩.
٨٢٠.              }
٨٢١.          }
٨٢٢.          public void btnThrotteldown_PreviewMouseDown(object se
      nder, MouseButtonEventArgs e)
٨٢٣.          {
٨٢٤.              if (throttel > 0 && throttel <= 255)
٨٢٥.              {
٨٢٦.                  throttel--;
٨٢٧.              }
٨٢٨.              pBThrottel.Value = throttel;
٨٢٩.              tbthrottel.Text = throttel.ToString();
٨٣٠.              if (keyboard_selected == true && connect_state ==
      true)
٨٣١.              {
٨٣٢.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٨٣٣.
٨٣٤.              }
٨٣٥.
٨٣٦.          }
٨٣٧.          public void btnYawup_PreviewMouseLeftButtonDown(object
       sender, MouseButtonEventArgs e)
٨٣٨.          {
٨٣٩.              if (yaw >= 127 && yaw < 255)
٨٤٠.              {
٨٤١.                  yaw++;
٨٤٢.              }
٨٤٣.              pBYawRight.Value = yaw;
٨٤٤.              tbyaw.Text = yaw.ToString();
٨٤٥.              if (keyboard_selected == true && connect_state ==
      true)
٨٤٦.              {
```

```
                send_jsk_parameter(firstchar, throttel, yaw, p
itch, roll);

            }
        }
        public void btnYawup_PreviewMouseLeftButtonUp(object s
ender, MouseButtonEventArgs e)
        {
            yaw = 127;
            pBYawRight.Value = yaw;
            tbyaw.Text = yaw.ToString();
            if (keyboard_selected == true && connect_state ==
true)
            {
                send_jsk_parameter(firstchar, throttel, yaw, p
itch, roll);

            }
        }
        public void btnYawdown_PreviewMouseLeftButtonDown(obje
ct sender, MouseButtonEventArgs e)
        {
            if (yaw <= 127 && yaw > 0)
            {
                yaw--;
            }
            pBYawLeft.Value = 127 - yaw;
            tbyaw.Text = yaw.ToString();
            if (keyboard_selected == true && connect_state ==
true)
            {
                send_jsk_parameter(firstchar, throttel, yaw, p
itch, roll);

            }
        }
        public void btnYawdown_PreviewMouseLeftButtonUp(object
 sender, MouseButtonEventArgs e)
        {
            yaw = 127;
            pBYawLeft.Value = 127 - yaw;
            tbyaw.Text = yaw.ToString();
            if (keyboard_selected == true && connect_state ==
true)
            {
                send_jsk_parameter(firstchar, throttel, yaw, p
itch, roll);

            }
```

```
٨٨٦.          }
٨٨٧.          public void btnPitchup_PreviewMouseLeftButtonDown(obje
      ct sender, MouseButtonEventArgs e)
٨٨٨.          {
٨٨٩.              if (pitch >= 127 && pitch < 255)
٨٩٠.              {
٨٩١.                  pitch++;
٨٩٢.              }
٨٩٣.              pBPitchRight.Value = pitch;
٨٩٤.              tbpitch.Text = pitch.ToString();
٨٩٥.              if (keyboard_selected == true && connect_state ==
      true)
٨٩٦.              {
٨٩٧.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٨٩٨.
٨٩٩.              }
٩٠٠.          }
٩٠١.          public void btnPitchup_PreviewMouseLeftButtonUp(object
       sender, MouseButtonEventArgs e)
٩٠٢.          {
٩٠٣.              pitch = 127;
٩٠٤.              pBPitchRight.Value = pitch;
٩٠٥.              tbpitch.Text = pitch.ToString();
٩٠٦.              if (keyboard_selected == true && connect_state ==
      true)
٩٠٧.              {
٩٠٨.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٩٠٩.
٩١٠.              }
٩١١.          }
٩١٢.          public void btnPitchdown_PreviewMouseLeftButtonDown(ob
      ject sender, MouseButtonEventArgs e)
٩١٣.          {
٩١٤.              if (pitch <= 127 && pitch > 0)
٩١٥.              {
٩١٦.                  pitch--;
٩١٧.              }
٩١٨.              pBPitchLeft.Value = 127 - pitch;
٩١٩.              tbpitch.Text = pitch.ToString();
٩٢٠.              if (keyboard_selected == true && connect_state ==
      true)
٩٢١.              {
٩٢٢.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٩٢٣.
٩٢٤.              }
٩٢٥.          }
```

```
٩٢٦.          public void btnPitchdown_PreviewMouseLeftButtonUp(obje
      ct sender, MouseButtonEventArgs e)
٩٢٧.          {
٩٢٨.              pitch = 127;
٩٢٩.              pBPitchLeft.Value = 127 - pitch;
٩٣٠.              tbpitch.Text = pitch.ToString();
٩٣١.              if (keyboard_selected == true && connect_state ==
      true)
٩٣٢.              {
٩٣٣.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٩٣٤.
٩٣٥.              }
٩٣٦.          }
٩٣٧.          public void btnRollup_PreviewMouseLeftButtonDown(objec
      t sender, MouseButtonEventArgs e)
٩٣٨.          {
٩٣٩.              if (roll >= 127 && roll < 255)
٩٤٠.              {
٩٤١.                  roll++;
٩٤٢.              }
٩٤٣.              pBRollRight.Value = roll;
٩٤٤.              tbroll.Text = roll.ToString();
٩٤٥.              if (keyboard_selected == true && connect_state ==
      true)
٩٤٦.              {
٩٤٧.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٩٤٨.
٩٤٩.              }
٩٥٠.          }
٩٥١.          public void btnRollup_PreviewMouseLeftButtonUp(object
      sender, MouseButtonEventArgs e)
٩٥٢.          {
٩٥٣.              roll = 127;
٩٥٤.              pBRollRight.Value = roll;
٩٥٥.              tbroll.Text = roll.ToString();
٩٥٦.              if (keyboard_selected == true && connect_state ==
      true)
٩٥٧.              {
٩٥٨.                  send_jsk_parameter(firstchar, throttel, yaw, p
      itch, roll);
٩٥٩.
٩٦٠.              }
٩٦١.          }
٩٦٢.          public void btnRolldown_PreviewMouseLeftButtonDown(obj
      ect sender, MouseButtonEventArgs e)
٩٦٣.          {
٩٦٤.              if (roll <= 127 && roll > 0)
```

```
                {
                    roll--;
                }
                pBRollLeft.Value = 127 - roll;
                tbroll.Text = roll.ToString();
                if (keyboard_selected == true && connect_state == true)
                {
                    send_jsk_parameter(firstchar, throttel, yaw, pitch, roll);

                }
            }
            public void btnRolldown_PreviewMouseLeftButtonUp(object sender, MouseButtonEventArgs e)
            {
                roll = 127;
                pBRollLeft.Value = 127 - roll;
                tbroll.Text = roll.ToString();
                if (keyboard_selected == true && connect_state == true)
                {
                    send_jsk_parameter(firstchar, throttel, yaw, pitch, roll);

                }
            }
            private void btnKeyboardSelect_Click(object sender, RoutedEventArgs e)
            {
                if (keyboard_selected == false && serial.IsOpen)
                {
                    keyboard_selected = true;
                    btnKeyboardSelect.Content = "Turn off";
                    gridKeyboard.IsEnabled = true;
                    JSK_state = false;
                    gridJSKcontrol.IsEnabled = false;
                    btnStartJSK.Content = "Start";
                    tbfirstchar.Text = firstchar.ToString();
                    tbthrottel.Text = throttel.ToString();
                    tbpitch.Text = pitch.ToString();
                    tbroll.Text = roll.ToString();
                    tbyaw.Text = yaw.ToString();

                }
                else if (keyboard_selected == true)
                {
                    keyboard_selected = false;
                    btnKeyboardSelect.Content = "Use Keyboard";
```

```
١٠٠٨.                   gridKeyboard.IsEnabled = false;
١٠٠٩.                   reset_keyboard();
١٠١٠.                   tbfirstchar.Text = "";
١٠١١.                   tbthrottel.Text = "";
١٠١٢.                   tbpitch.Text = "";
١٠١٣.                   tbroll.Text = "";
١٠١٤.                   tbyaw.Text = "";
١٠١٥.
١٠١٦.               }
١٠١٧.           }
١٠١٨.           private void btnGoto_Click(object sender, RoutedEventA
      rgs e)
١٠١٩.           {
١٠٢٠.               try
١٠٢١.               {
١٠٢٢.                   float lat = float.Parse(txtLatitude.Text, Syst
      em.Globalization.CultureInfo.InvariantCulture);
١٠٢٣.                   float lng = float.Parse(txtLongitude.Text, Sys
      tem.Globalization.CultureInfo.InvariantCulture);
١٠٢٤.                   PointLatLng selectpoint2 = new PointLatLng(lat
      , lng);
١٠٢٥.                   mapControl.Position = new PointLatLng(lat, lng
      );
١٠٢٦.                   mapControl.Zoom = sliderZoom.Value;
١٠٢٧.
١٠٢٨.                   GMapMarker marker = new GMapMarker(selectpoint
      2);
١٠٢٩.                   Ellipse temp = new Ellipse();
١٠٣٠.                   temp.Margin = new Thickness(-10);
١٠٣١.                   temp.Height = 10;
١٠٣٢.                   temp.Width = 10;
١٠٣٣.                   if (int.Parse(spDT.txtSignalStrength.Text) >=
      10 && int.Parse(spDT.txtSignalStrength.Text) <= 65) { temp.Fil
      l = Brushes.Blue; }
١٠٣٤.                   else if (int.Parse(spDT.txtSignalStrength.Text
      ) > 65 && int.Parse(spDT.txtSignalStrength.Text) <= 75) { temp
      .Fill = Brushes.GreenYellow; }
١٠٣٥.                   else if (int.Parse(spDT.txtSignalStrength.Text
      ) > 75 && int.Parse(spDT.txtSignalStrength.Text) <= 85) { temp
      .Fill = Brushes.Green; }
١٠٣٦.                   else if (int.Parse(spDT.txtSignalStrength.Text
      ) > 85 && int.Parse(spDT.txtSignalStrength.Text) <= 95) { temp
      .Fill = Brushes.Yellow; }
١٠٣٧.                   else if (int.Parse(spDT.txtSignalStrength.Text
      ) > 95 && int.Parse(spDT.txtSignalStrength.Text) <= 115) { tem
      p.Fill = Brushes.Red; }
١٠٣٨.
١٠٣٩.                   marker.Shape = temp;
١٠٤٠.
```

```
                    mapControl.Markers.Add(marker);

                }
                catch (Exception ex) { MessageBox.Show(ex.ToString
        ()); }
            }
            private void btnStartMotor_Click(object sender, Routed
        EventArgs e)
            {

                if (connect_state == true)
                {
                    if (MessageBox.Show("Are you sure ? if yes mot
        ors will start :)", "Question", MessageBoxButton.YesNo, Messag
        eBoxImage.Warning) == MessageBoxResult.Yes)
                    {
                        SerialCmdSend("S");
                        btnKeyboardSelect.IsEnabled = true;

                    }
                    else
                    {

                    }


                }
            }
            private void btnStopMotor_Click(object sender, RoutedE
        ventArgs e)
            {
                if (connect_state == true)
                {
                    if (MessageBox.Show("Are you sure ? if yes mot
        ors will Stop :)", "Question", MessageBoxButton.YesNo, Message
        BoxImage.Warning) == MessageBoxResult.Yes)
                    {
                        if (throttel <= 100)
                        {
                            SerialCmdSend("P");
                            if (keyboard_selected) { btnKeyboardSe
        lect_Click(null, null); }
                            btnKeyboardSelect.IsEnabled = false;
                            reset_keyboard();
                        }
                        else
                        {
```

```
                        MessageBox.Show("Sorry Throttel is more than 40% of max And it may cause damage.you can use Landing Mode ", "Question", MessageBoxButton.OK, MessageBoxImage.Warning);
                    }
                }
                else
                {

                }
            }
        }
        private void btnLandingMode_Click(object sender, RoutedEventArgs e)
        {
            if (connect_state == true)
            {
                if (MessageBox.Show("Are you sure ? if yes your Quadcopter will enter landing mode :)", "Question", MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)
                {
                    SerialCmdSend("L");
                    reset_keyboard();

                }
                else
                {

                }
            }
        }

        private void btnClearErrortxt_Click(object sender, RoutedEventArgs e)
        {
            txtError.Text = "";
        }

        private void btnStartReceiveData_Click(object sender, RoutedEventArgs e)
        {
            if (connect_state == true && serial.IsOpen)
            {
                serial.Write("T");
            }
        }

```

```
١١٢٠.         private void mapControl_PreviewMouseWheel(object sende
      r, MouseWheelEventArgs e)
١١٢١.         {
١١٢٢.             sliderZoom.Value = mapControl.Zoom;
١١٢٣.         }
١١٢٤.
١١٢٥.         private void MenuItem_Click(object sender, RoutedEvent
      Args e)
١١٢٦.         {
١١٢٧.             windowHelpAbout window = new windowHelpAbout();
١١٢٨.             var result = window.ShowDialog();
١١٢٩.             //  window.Show();
١١٣٠.             //  this.IsEnabled=false;
١١٣١.         }
١١٣٢.
١١٣٣.         private void btnmenuDataBase_Click(object sender, Rout
      edEventArgs e)
١١٣٤.         {
١١٣٥.             windowDataBaseGrid win = new windowDataBaseGrid();

١١٣٦.             win.Show();
١١٣٧.         }
١١٣٨.
١١٣٩.         private void btnFoucsArea_Click(object sender, RoutedE
      ventArgs e)
١١٤٠.         {
١١٤١.             mapControl.SetPositionByKeywords(txtFoucsArea.Text
      );
١١٤٢.         }
١١٤٣.
١١٤٤.         private void btnClearMark_Click(object sender, RoutedE
      ventArgs e)
١١٤٥.         {
١١٤٦.             mapControl.Markers.Clear();
١١٤٧.         }
١١٤٨.
١١٤٩.         private void btnMonileDataUpdate_Click(object sender,
      RoutedEventArgs e)
١١٥٠.         {
١١٥١.             if (connect_state == true)
١١٥٢.             {
١١٥٣.                 SerialCmdSend("N");
١١٥٤.             }
١١٥٥.         }
١١٥٦.
١١٥٧.     }
١١٥٨. }
```

Controller

```
١.  using System;
٢.  using System.Collections.Generic;
٣.  using System.Linq;
٤.  using System.Text;
٥.  using System.Threading.Tasks;
٦.  using System.Windows;
٧.  using System.Windows.Controls;
٨.  using System.Windows.Data;
٩.  using System.Windows.Documents;
١٠. using System.Windows.Input;
١١. using System.Windows.Media;
١٢. using System.Windows.Media.Imaging;
١٣. using System.Windows.Navigation;
١٤. using System.Windows.Shapes;
١٥. using MySql.Data.MySqlClient;
١٦.
١٧. namespace SerialPortControl
١٨. {
١٩.     /// <summary>
٢٠.     /// Interaction logic for controllerDriveTest.xaml
٢١.     /// </summary>
٢٢.     public partial class controllerDriveTest : UserControl
٢٣.     {
٢٤.         public controllerDriveTest()
٢٥.         {
٢٦.             InitializeComponent();
٢٧.             get_Period();
٢٨.         }
٢٩.
٣٠.
٣١.         #region Function
٣٢.         void get_Period()
٣٣.         {
٣٤.             txtPeriod.Items.Clear();
٣٥.             string[] baud = { "1", "5", "10", "100", "1000", "10000", "100000", "1
    000000" };
٣٦.             foreach (string Rate in baud)
٣٧.             {
٣٨.                 txtPeriod.Items.Add(Rate);
٣٩.             }
٤٠.         }
٤١.
٤٢.         public string DateFormat
٤٣.         {
٤٤.             get
٤٥.             {
٤٦.                 string Date = txtDate.Text;
٤٧.
٤٨.                 string[] str = Date.Split('/');
٤٩.                 string day = str[1];
٥٠.                 string month = str[0];
```

```
                string year = str[2];
                string FullDate = year + "-" + month + "-" + day;
                return FullDate;

            }
        }

        #endregion

        #region variable
        bool CreatTable = false;
        string MyConnection = "datasource=localhost;port=3306;username=root;passwo
rd=Bikamak15";
        System.Windows.Threading.DispatcherTimer dispatcherTimer = new System.Wind
ows.Threading.DispatcherTimer();

        #endregion

        #region Timer
        private void dispatcherTimer_Tick(object sender, EventArgs e)
        {
            if (CreatTable == true)
            {
                string DataToInsert = "INSERT INTO `quadcopterdrivetest`.`" + txtA
reaName.Text + "_" + DateFormat + "`" +
                                      "(`operatorname`, `mcc`, `SIMstate`, `roamin
g`," +
                                      "`networktype`, `phonetype`, `imei`, `imsi`,
 " +
                                      "`longitude`, `latitude`, `cellid`," +
                                      "`locationareacode`, `signalstrength`," +
                                      "`biterrrorrate`, `level`, `rssi`," +
                                      "`ecio`, `snr`)" +
                                      "VALUES" +
                                      "('" + txtOperatorName.Text + "', '" + txtMc
c.Text + "', '" + txtSimState.Text +
                                      "', '" + txtRoaming.Text + "', '" + txtNetwo
rkType.Text + "', '" + txtPhoneType.Text +
                                      "', '" + txtImei.Text + "', '" + txtImsi.Tex
t + "', '" + txtLongitude.Text +
                                      "', '" + txtLatitude.Text + "', '" + txtCell
Id.Text + "', '" + txtLocationAreaCode.Text +
                                      "', '" + txtSignalStrength.Text + "', '" + t
xtBitErrorRate.Text + "', '" + txtLevel.Text +
                                      "', '" + txtRssi.Text + "', '" + txtEcio.Tex
t + "', '" + txtSNR.Text + "');";
                MySqlConnection myconn = new MySqlConnection(MyConnection);
                MySqlCommand cmdDataBase = new MySqlCommand(DataToInsert, myconn);

                MySqlDataReader MyReader;
                try
                {
                    myconn.Open();
                    MyReader = cmdDataBase.ExecuteReader();

                }


                catch (Exception exp)
                {
                    MessageBox.Show(exp.ToString());
                    }
                }

            }
```

```
١٠٤.
١٠٥.                #endregion
١٠٦.                private void btnNewTable_Click(object sender, RoutedEventArgs e)
١٠٧.                {
١٠٨.                    if (CreatTable == false)
١٠٩.                    {
١١٠.                        CreatTable = true;
١١١.                        gridNewTable.IsEnabled = false;
١١٢.                        gridDriveTestData.IsEnabled = true;
١١٣.                        btnNewTable.Content = "Stop Measurement";
١١٤.                        dispatcherTimer.Tick += dispatcherTimer_Tick;
١١٥.                        dispatcherTimer.Interval = new TimeSpan(0, 0, 0, 0, int.Par
    se(txtPeriod.Text));
١١٦.                        dispatcherTimer.Start();
١١٧.                        string DataToInsert = " CREATE TABLE IF NOT EXISTS `quadcop
    terdrivetest`.`" + txtAreaName.Text + "_" + DateFormat + "`" +
١١٨.                                              " (`Id` INT NOT NULL AUTO_INCREMENT
    ," +
١١٩.                                              "`operatorname` VARCHAR(45) NULL,"
    +
١٢٠.                                              "`mcc` VARCHAR(45) NULL," +
١٢١.                                              "`SIMstate` VARCHAR(45) NULL," +
١٢٢.                                              "`roaming` VARCHAR(45) NULL," +
١٢٣.                                              "`networktype` VARCHAR(45) NULL," +

١٢٤.                                              "`phonetype` VARCHAR(45) NULL," +
١٢٥.                                              "`imei` VARCHAR(45) NULL," +
١٢٦.                                              "`imsi` VARCHAR(45) NULL," +
١٢٧.                                              "`longitude` VARCHAR(45) NULL," +
١٢٨.                                              "`latitude` VARCHAR(45) NULL," +
١٢٩.                                              "`cellid` VARCHAR(45) NULL," +
١٣٠.                                              "`locationareacode` VARCHAR(45) NUL
    L," +
١٣١.                                              "`signalstrength` VARCHAR(45) NULL,
    " +
١٣٢.                                              "`biterrrorrate` VARCHAR(45) NULL,"
     +
١٣٣.                                              "`level` VARCHAR(45) NULL," +
١٣٤.                                              "`rssi` VARCHAR(45) NULL," +
١٣٥.                                              "`ecio` VARCHAR(45) NULL," +
١٣٦.                                              "`snr` VARCHAR(45) NULL," +
١٣٧.                                              "PRIMARY KEY (`Id`));";
١٣٨.                        MySqlConnection myconn = new MySqlConnection(MyConnection);

١٣٩.                        MySqlCommand cmdDataBase = new MySqlCommand(DataToInsert, m
    yconn);
١٤٠.                        MySqlDataReader MyReader;
١٤١.                        try
١٤٢.                        {
١٤٣.                            myconn.Open();
١٤٤.                            MyReader = cmdDataBase.ExecuteReader();
١٤٥.                            CreatTable = true;
١٤٦.                        }
١٤٧.
١٤٨.
١٤٩.                        catch (Exception exp)
١٥٠.                        {
١٥١.                            MessageBox.Show(exp.ToString());
١٥٢.                        }
١٥٣.
١٥٤.
١٥٥.                    }
١٥٦.                    else if (CreatTable == true)
١٥٧.                    {
١٥٨.                        CreatTable = false;
```

```
                        gridNewTable.IsEnabled = true;
                        gridDriveTestData.IsEnabled = false;
                        btnNewTable.Content = "New Drive Test Measurement";
                    }

                }

                private void txtPeriod_SelectionChanged(object sender, SelectionCha
    ngedEventArgs e)
                {

                }
            }
        }
```

Data grid

```
١. using System;
٢. using System.Collections.Generic;
٣. using System.Linq;
٤. using System.Text;
٥. using System.Threading.Tasks;
٦. using System.Windows;
٧. using System.Windows.Controls;
٨. using System.Windows.Data;
٩. using System.Windows.Documents;
١٠. using System.Windows.Input;
١١. using System.Windows.Media;
١٢. using System.Windows.Media.Imaging;
١٣. using System.Windows.Shapes;
١٤. using MySql.Data.MySqlClient;
١٥. using System.Data;
١٦. using Microsoft.Win32;
١٧. using System.Web;
١٨. using GMap.NET.MapProviders;
١٩. using GMap.NET;
٢٠. using GMap.NET.WindowsPresentation;
٢١.
٢٢.
٢٣. namespace SerialPortControl
٢٤. {
٢٥.     /// <summary>
٢٦.     /// Interaction logic for windowDataBaseGrid.xaml
٢٧.     /// </summary>
٢٨.     public partial class windowDataBaseGrid : Window
٢٩.     {
٣٠.
٣١.         string Table_Name;
٣٢.         bool StartDataGrid = false;
٣٣.         public void FillGrid()
٣٤.         {
٣٥.             string MyConnection = "datasource=localhost;port=3306;username=root;pa
    ssword=Bikamak15";
٣٦.             string sql = "SELECT * FROM quadcopterdrivetest.`" + Table_Name + "`;"
    ;
٣٧.
٣٨.             MySqlConnection connection = new MySqlConnection(MyConnection);
٣٩.             MySqlCommand cmdSel = new MySqlCommand(sql, connection);
٤٠.             DataTable dt = new DataTable();
٤١.             MySqlDataAdapter da = new MySqlDataAdapter(cmdSel);
٤٢.             da.Fill(dt);
٤٣.             DataGrid.DataContext = dt;
٤٤.         }
٤٥.
٤٦.         void FillListBox()
٤٧.         {
٤٨.             lstTableName.Items.Clear();
٤٩.             lstTableName.SelectedIndex = 0;
٥٠.             string MyConnection = "datasource=localhost;port=3306;username=root;pa
    ssword=Bikamak15";
٥١.             string DataToInsert = "SHOW TABLES IN quadcopterdrivetest;";
٥٢.             MySqlConnection myconn = new MySqlConnection(MyConnection);
٥٣.             MySqlCommand cmdDataBase = new MySqlCommand(DataToInsert, myconn);
٥٤.             try
٥٥.             {
٥٦.                 MySqlDataReader MyReader;
٥٧.
٥٨.                 myconn.Open();
```

```
                MyReader = cmdDataBase.ExecuteReader();
                while (MyReader.Read())
                {
                    string TableName = MyReader.GetString(0);
                    lstTableName.Items.Add(TableName);
                    StartDataGrid = true;
                }
                Table_Name = lstTableName.SelectedItem.ToString();

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }

        }

        void DropTable()
        {
            string MyConnection = "datasource=localhost;port=3306;username=root;password=Bikamak15";
            string DataToInsert = "DROP TABLE `quadcopterdrivetest`.`" + Table_Name + "`;";
            MySqlConnection myconn = new MySqlConnection(MyConnection);
            MySqlCommand cmdDataBase = new MySqlCommand(DataToInsert, myconn);
            try
            {
                MySqlDataReader MyReader;

                myconn.Open();
                MyReader = cmdDataBase.ExecuteReader();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }

        void point_mark_map(string Latitude ,string Longitude, string SignalStrength)
        {
            try
            {
                float lat = float.Parse(Latitude, System.Globalization.CultureInfo.InvariantCulture);
                float lng = float.Parse(Longitude, System.Globalization.CultureInfo.InvariantCulture);
                PointLatLng selectpoint2 = new PointLatLng(lat, lng);
                mapControl.Position = new PointLatLng(lat, lng);


                GMapMarker marker = new GMapMarker(selectpoint2);
                Ellipse temp = new Ellipse();
                temp.Margin = new Thickness(-10);
                temp.Height = 10;
                temp.Width = 10;
                if (int.Parse(SignalStrength) >= 10 && int.Parse(SignalStrength) <= 65) { temp.Fill = Brushes.Blue; }
                else if (int.Parse(SignalStrength) > 65 && int.Parse(SignalStrength) <= 75) { temp.Fill = Brushes.GreenYellow; }
                else if (int.Parse(SignalStrength) > 75 && int.Parse(SignalStrength) <= 85) { temp.Fill = Brushes.Green; }
                else if (int.Parse(SignalStrength) > 85 && int.Parse(SignalStrength) <= 95) { temp.Fill = Brushes.Yellow; }
```

```
                        else if (int.Parse(SignalStrength) > 95 && int.Parse(Signal
    Strength) <= 115) { temp.Fill = Brushes.Red; }

                        marker.Shape = temp;

                        mapControl.Markers.Add(marker);

                    }
                    catch (Exception ex) { MessageBox.Show(ex.ToString()); }
                }
                private void ExportToExcel()
                {
                    Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFi
    leDialog();
                    dlg.FileName = "Excel Sheet"; // Default file name
                    dlg.DefaultExt = ".xls"; // Default file extension
                    dlg.Filter = "The Excel sheet (.xls)|*.xls"; // Filter files by
     extension

                    // Show save file dialog box
                    Nullable<bool> result = dlg.ShowDialog();

                    // Process save file dialog box results
                    if (result == true)
                    {
                        // Save document
                        string filename = dlg.FileName;
                        DataGrid.SelectAllCells();
                        DataGrid.ClipboardCopyMode = DataGridClipboardCopyMode.Incl
    udeHeader;
                        ApplicationCommands.Copy.Execute(null, DataGrid);
                        String resultat = (string)Clipboard.GetData(DataFormats.Com
    maSeparatedValue);
                        String result2 = (string)Clipboard.GetData(DataFormats.Text
    );
                        DataGrid.UnselectAllCells();
                        System.IO.StreamWriter file = new System.IO.StreamWriter(@f
    ilename);
                        file.WriteLine(result2.Replace(',', ' '));
                        file.Close();
                        MessageBox.Show("Exporting DataGrid data to Excel file crea
    ted");

                    }


                }
                public windowDataBaseGrid()
                {
                    InitializeComponent();
                    FillListBox();
                    System.Windows.Threading.DispatcherTimer dispatcherTimer = new
    System.Windows.Threading.DispatcherTimer();
                    dispatcherTimer.Tick += dispatcherTimer_Tick;
                    dispatcherTimer.Interval = new TimeSpan(0, 0, 0, 5);
                    dispatcherTimer.Start();
                    StartDataGrid = true;
                }


                private void dispatcherTimer_Tick(object sender, EventArgs e)
                {
                    if (StartDataGrid == true && lstTableName.Items.Count > 0)
                    {
                        FillGrid();
```

```
                }
            }

            private void btnRefreshDataBase_Click(object sender, RoutedEventArg
s e)
            {
                if (StartDataGrid == true && lstTableName.Items.Count > 0)
                {
                    FillListBox();
                    FillGrid();
                }
            }

            private void lstTableName_SelectionChanged(object sender, Selection
ChangedEventArgs e)
            {
                if (StartDataGrid == true && lstTableName.Items.Count > 0)
                {
                    Table_Name = lstTableName.SelectedItem.ToString();
                    FillGrid();
                }
            }

            private void btnDeleteTable_Click(object sender, RoutedEventArgs e)

            {
                if (lstTableName.Items.Count >1)
                {
                    StartDataGrid = false;
                    DropTable();
                    FillListBox();
                    Table_Name = lstTableName.SelectedItem.ToString();
                    StartDataGrid = true;
                    FillGrid();

                }
                else if (lstTableName.Items.Count == 1)
                {

                    DropTable();
                    StartDataGrid = false;

                }



            }

            private void btnExportToExcel_Click(object sender, RoutedEventArgs
e)
            {
                ExportToExcel();
            }

            private void btnViewMap_PreviewMouseLeftButtonDown(object sender, M
ouseButtonEventArgs e)
            {
                mapControl.Visibility=Visibility.Visible;
                mapControl.MapProvider = GoogleMapProvider.Instance;
                mapControl.Position = new PointLatLng(29.3068032975785, 30.8506
90305233);
                mapControl.Zoom = 15;

```

```
                    string MyConnection = "datasource=localhost;port=3306;username=root;password=Bikamak15";
                    string DataToInsert = "SELECT * FROM quadcopterdrivetest.`" + Table_Name + "`;";
                    MySqlConnection myconn = new MySqlConnection(MyConnection);
                    MySqlCommand cmdDataBase = new MySqlCommand(DataToInsert, mycon);
                    try
                    {
                        MySqlDataReader MyReader;

                        myconn.Open();
                        MyReader = cmdDataBase.ExecuteReader();
                        while (MyReader.Read())
                        {
                            string longitude = MyReader.GetString("longitude");
                            string latitude = MyReader.GetString("latitude");
                            string signalstrength = MyReader.GetString("signalstrength");
                            signalstrength = signalstrength.Replace("-", string.Empty);
                            if(longitude!="" && latitude!="" && signalstrength!="")
                            {
                                point_mark_map(latitude, longitude, signalstrength);

                            }

                        }

                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show(ex.ToString());
                    }

                }

                private void btnViewMap_PreviewMouseLeftButtonUp(object sender, MouseButtonEventArgs e)
                {
                    mapControl.Markers.Clear();
                    mapControl.Visibility = Visibility.Hidden;

                }

                private void btnViewMap_Click(object sender, RoutedEventArgs e)
                {

                }
            }
        }
```

Android App Code

Main activity code

```
١. package com.example.bassem.final_project;
٢.
٣. import android.app.PendingIntent;
٤. import android.content.BroadcastReceiver;
٥. import android.content.Context;
٦. import android.content.Intent;
٧. import android.content.IntentFilter;
٨. import android.hardware.usb.UsbDevice;
٩. import android.hardware.usb.UsbDeviceConnection;
١٠. import android.hardware.usb.UsbManager;
١١. import android.os.Bundle;
١٢. import android.os.Handler;
١٣. import android.support.design.widget.FloatingActionButton;
١٤. import android.support.design.widget.Snackbar;
١٥. import android.support.v٧.app.AppCompatActivity;
١٦. import android.support.v٧.widget.Toolbar;
١٧. import android.util.Log;
١٨. import android.view.View;
١٩. import android.view.Menu;
٢٠. import android.view.MenuItem;
٢١. import android.widget.Button;
٢٢. import android.widget.TextView;
٢٣.
٢٤. import com.felhr.usbserial.UsbSerialDevice;
٢٥. import com.felhr.usbserial.UsbSerialInterface;
٢٦.
٢٧. import java.io.UnsupportedEncodingException;
٢٨. import java.util.HashMap;
٢٩. import java.util.Map;
٣٠.
٣١. public class MainActivity extends AppCompatActivity implements View.OnClickListener
     {
٣٢.
٣٣.     Button GSM , CDMA , INFO , USBHOST;
٣٤.
٣٥.
٣٦.     // usbserial
٣٧.
٣٨.     public final String ACTION_USB_PERMISSION = "com.hariharan.arduinousb.USB_PERMI
    SSION";
٣٩.     UsbManager usbManager;
٤٠.     UsbDevice device;
٤١.     UsbSerialDevice serialPort;
٤٢.     UsbDeviceConnection connection;
٤٣.
٤٤.     UsbSerialInterface.UsbReadCallback mCallback = new UsbSerialInterface.UsbReadCa
    llback()   //Defining a Callback which triggers whenever data is read.
٤٥.     {
٤٦.         @Override
٤٧.         public void onReceivedData(byte[] arg٠) {
٤٨.             String data = null;
٤٩.             try {
٥٠.                 data = new String(arg٠, "UTF-٨");
٥١.                 data.concat("/n");
٥٢.                // tvAppend(usb, data);
٥٣.             } catch (UnsupportedEncodingException e) {
٥٤.                 e.printStackTrace();
٥٥.             }
```

```
٥٦.
٥٧.
٥٨.         }
٥٩.     };
٦٠.
٦١.     private final BroadcastReceiver broadcastReceiver = new BroadcastReceiver() { //Broadcast Receiver to automatically start and stop the Serial connection.
٦٢.         @Override
٦٣.         public void onReceive(Context context, Intent intent) {
٦٤.             if (intent.getAction().equals(ACTION_USB_PERMISSION)) {
٦٥.                 boolean granted = intent.getExtras().getBoolean(UsbManager.EXTRA_PERMISSION_GRANTED);
٦٦.                 if (granted) {
٦٧.                     connection = usbManager.openDevice(device);
٦٨.                     serialPort = UsbSerialDevice.createUsbSerialDevice(device, connection);
٦٩.                     if (serialPort != null) {
٧٠.                         if (serialPort.open()) {
٧١.                             serialPort.setBaudRate(٩٦٠٠);
٧٢.                             serialPort.setDataBits(UsbSerialInterface.DATA_BITS_٨);
٧٣.                             serialPort.setStopBits(UsbSerialInterface.STOP_BITS_١);
٧٤.                             serialPort.setParity(UsbSerialInterface.PARITY_NONE);
٧٥.                             serialPort.setFlowControl(UsbSerialInterface.FLOW_CONTROL_OFF);
٧٦.                             serialPort.read(mCallback);
٧٧.                         } else {
٧٨.                             Log.d("SERIAL", "PORT NOT OPEN");
٧٩.                         }
٨٠.                     } else {
٨١.                         Log.d("SERIAL", "PORT IS NULL");
٨٢.                     }
٨٣.                 } else {
٨٤.                     Log.d("SERIAL", "PERM NOT GRANTED");
٨٥.                 }
٨٦.             } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_ATTACHED)) {
٨٧.                 start();
٨٨.             } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_DETACHED)) {
٨٩.                 serialPort.close();
٩٠.             }
٩١.         }
٩٢.     };
٩٣.
٩٤.
٩٥.     @Override
٩٦.     protected void onCreate(Bundle savedInstanceState) {
٩٧.         super.onCreate(savedInstanceState);
٩٨.         setContentView(R.layout.activity_main);
٩٩.
١٠٠.
١٠١.
١٠٢.         GSM = (Button)findViewById(R.id.Main_GSM);     //casting as a button
١٠٣.         CDMA = (Button)findViewById(R.id.Main_CDMA);
١٠٤.         INFO = (Button)findViewById(R.id.Main_INFO);
١٠٥.         USBHOST = (Button)findViewById(R.id.MAIN_OTG);
١٠٦.
١٠٧.
١٠٨.         GSM.setOnClickListener(this);
١٠٩.         CDMA.setOnClickListener(this);
١١٠.         INFO.setOnClickListener(this);
١١١.         USBHOST.setOnClickListener(this);
١١٢.
```

```
١١٣.
١١٤.        usbManager = (UsbManager) getSystemService(this.USB_SERVICE);
١١٥.        IntentFilter filter = new IntentFilter();
١١٦.        filter.addAction(ACTION_USB_PERMISSION);
١١٧.        filter.addAction(UsbManager.ACTION_USB_DEVICE_ATTACHED);
١١٨.        filter.addAction(UsbManager.ACTION_USB_DEVICE_DETACHED);
١١٩.        registerReceiver(broadcastReceiver, filter);
١٢٠.
١٢١.        //step ١
١٢٢.        // start serial
١٢٣.        USBHOST.performClick();
١٢٤.
١٢٥.        //step ٢
١٢٦.        //sending general info.
١٢٧.        final Handler handler = new Handler();
١٢٨.        handler.postDelayed(new Runnable() {
١٢٩.            @Override
١٣٠.            public void run() {
١٣١.                // Do something after ٣.٥s = ٣٥٠٠ms
١٣٢.                INFO.performClick();
١٣٣.            }
١٣٤.        }, ٣٥٠٠);
١٣٥.
١٣٦.        //step٣
١٣٧.        //sending GSM info.
١٣٨.        final Handler handler١ = new Handler();
١٣٩.        handler١.postDelayed(new Runnable() {
١٤٠.            @Override
١٤١.            public void run() {
١٤٢.                // Do something after ٨.٥s = ٨٥٠٠ms
١٤٣.                start();
١٤٤.                GSM.performClick();
١٤٥.            }
١٤٦.        }, ٨٥٠٠);
١٤٧.
١٤٨.    }
١٤٩.
١٥٠.    public void start()
١٥١.    {
١٥٢.        HashMap<String, UsbDevice> usbDevices = usbManager.getDeviceList();
١٥٣.        if (!usbDevices.isEmpty()) {
١٥٤.            boolean keep = true;
١٥٥.            for (Map.Entry<String, UsbDevice> entry : usbDevices.entrySet()) {
١٥٦.                device = entry.getValue();
١٥٧.                int deviceVID = device.getVendorId();
١٥٨.                if (deviceVID == ٠x٢٣٤١)               //Arduino Vendor ID
١٥٩.                {
١٦٠.
    PendingIntent pi = PendingIntent.getBroadcast(this, ٠, new Intent(ACTION_USB_PERMIS
    SION), ٠);
١٦١.                    usbManager.requestPermission(device, pi);
١٦٢.                    keep = false;
١٦٣.                } else {
١٦٤.                    connection = null;
١٦٥.                    device = null;
١٦٦.                }
١٦٧.
١٦٨.                if (!keep)
١٦٩.                    break;
١٧٠.            }
١٧١.        }
١٧٢.    }
١٧٣.
١٧٤.
١٧٥.
```

```
١٧٦.
١٧٧.     @Override
١٧٨.     public void onClick(View v)
١٧٩.     {
١٨٠.         if (v == GSM)
١٨١.         {
١٨٢.             Intent intent = new Intent(MainActivity.this,GSM.class);
١٨٣.             startActivity(intent);
١٨٤.         }
١٨٥.
١٨٦.         else if (v == CDMA)
١٨٧.         {
١٨٨.             Intent intent١ = new Intent(MainActivity.this,CDMA.class);
١٨٩.             startActivity(intent١);
١٩٠.         }
١٩١.
١٩٢.         else if (v == INFO)
١٩٣.         {
١٩٤.             Intent intent٢ = new Intent(MainActivity.this,INFO.class);
١٩٥.             startActivity(intent٢);
١٩٦.         }
١٩٧.
١٩٨.         else if (v == USBHOST)
١٩٩.         {
٢٠٠.             start();
٢٠١.         }
٢٠٢.     }
٢٠٣. }
```

Gsm activity code

```
١. package com.example.bassem.final_project;
٢.
٣. import android.app.PendingIntent;
٤. import android.content.BroadcastReceiver;
٥. import android.content.Context;
٦. import android.content.Intent;
٧. import android.content.IntentFilter;
٨. import android.hardware.usb.UsbDevice;
٩. import android.hardware.usb.UsbDeviceConnection;
١٠. import android.hardware.usb.UsbManager;
١١. import android.os.Bundle;
١٢. import android.support.v٧.app.AppCompatActivity;
١٣. import android.telephony.CellIdentityGsm;
١٤. import android.telephony.PhoneStateListener;
١٥. import android.telephony.SignalStrength;
١٦. import android.telephony.TelephonyManager;
١٧. import android.telephony.gsm.GsmCellLocation;
١٨. import android.util.Log;
١٩. import android.view.View;
٢٠. import android.widget.Button;
٢١. import android.widget.TextView;
٢٢. import android.widget.Toast;
٢٣. import android.os.Handler;
٢٤.
٢٥. import com.felhr.usbserial.UsbSerialDevice;
٢٦. import com.felhr.usbserial.UsbSerialInterface;
٢٧.
٢٨. import java.io.UnsupportedEncodingException;
٢٩. import java.util.HashMap;
٣٠. import java.util.Map;
٣١.
٣٢.
٣٣. public class GSM extends AppCompatActivity implements View.OnClickListener {
٣٤.
٣٥.         PhoneStateListener MPhoneStateListener ;
٣٦.         TelephonyManager Tm ;
٣٧.
٣٨.     // usbserial
٣٩.
٤٠.  public final String ACTION_USB_PERMISSION = "com.hariharan.arduinousb.USB_PERMISSI
    ON";
٤١.     UsbManager usbManager;
٤٢.     UsbDevice device;
٤٣.     UsbSerialDevice serialPort;
٤٤.     UsbDeviceConnection connection;
٤٥.
٤٦. UsbSerialInterface.UsbReadCallback mCallback = new UsbSerialInterface.UsbReadCallba
    ck()
٤٧.     {
٤٨.         @Override
٤٩.         public void onReceivedData(byte[] arg٠) {
٥٠.             String data = null;
٥١.             try {
٥٢.                 data = new String(arg٠, "UTF-٨");
٥٣.                 data.concat("/n");
٥٤.                 tvAppend(usb, data);
٥٥.             } catch (UnsupportedEncodingException e) {
٥٦.                 e.printStackTrace();
٥٧.             }
٥٨.
٥٩.
```

```
        }
    };
    private final BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            if (intent.getAction().equals(ACTION_USB_PERMISSION)) {
                boolean granted = intent.getExtras().getBoolean(UsbManager.EXTRA_PE
  RMISSION_GRANTED);
                if (granted) {
                    connection = usbManager.openDevice(device);
        serialPort = UsbSerialDevice.createUsbSerialDevice(device, connection);
                    if (serialPort != null) {
                        if (serialPort.open()) {

                            serialPort.setBaudRate(٩٦٠٠);
                           serialPort.setDataBits(UsbSerialInterface.DATA_BITS_٨);
                           serialPort.setStopBits(UsbSerialInterface.STOP_BITS_١);
                           serialPort.setParity(UsbSerialInterface.PARITY_NONE);
                    serialPort.setFlowControl(UsbSerialInterface.FLOW_CONTROL_OFF);
                            serialPort.read(mCallback);
                            tvAppend(usb, "Serial Connection Opened!\n");
                        } else {
                            Log.d("SERIAL", "PORT NOT OPEN");
                        }
                    } else {
                        Log.d("SERIAL", "PORT IS NULL");
                    }
                } else {
                    Log.d("SERIAL", "PERM NOT GRANTED");
                }
   } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_ATTACHED)) {
                start();
   } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_DETACHED)) {
                serialPort.close();
            }
        }
    };


    TextView SS , LE , usb , cellid , lac;
    Button  delete , data;
     int signalstr  , level;
     int CELLid , LAC;
     MyDBHandler dbHandler;

     @Override
     protected void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.activity_gsm);

         Tm = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
         SS = (TextView)findViewById(R.id.GSM_SS);
         LE = (TextView)findViewById(R.id.GSM_level);
         cellid = (TextView)findViewById(R.id.GSM_CELLID);
         lac = (TextView)findViewById(R.id.GSM_LAC);
         usb = (TextView)findViewById(R.id.GSM_USB);
         delete = (Button)findViewById(R.id.GSM_DELETE);
         data = (Button)findViewById(R.id.GSM_data);
         usbManager = (UsbManager) getSystemService(this.USB_SERVICE);


          data.setOnClickListener(this);
          delete.setOnClickListener(this);
```

```
١٢٣.
١٢٤.        dbHandler = new MyDBHandler(this);
١٢٥.        Log.i("Tag", "activity created");
١٢٦.
١٢٧.        TelephonyManager telephonyManager = (TelephonyManager) getSystemService(Co
    ntext.TELEPHONY_SERVICE);
١٢٨.
    GsmCellLocation cellLocation = (GsmCellLocation) telephonyManager.getCellLocation()
    ;
١٢٩.
١٣٠.        // Cell Id, LAC
١٣١.
١٣٢.        CELLid = cellLocation.getCid();
١٣٣.
١٣٤.        LAC = cellLocation.getLac();
١٣٥.
١٣٦.        IntentFilter filter = new IntentFilter();
١٣٧.        filter.addAction(ACTION_USB_PERMISSION);
١٣٨.        filter.addAction(UsbManager.ACTION_USB_DEVICE_ATTACHED);
١٣٩.        filter.addAction(UsbManager.ACTION_USB_DEVICE_DETACHED);
١٤٠.        registerReceiver(broadcastReceiver, filter);
١٤١.
١٤٢.        final int delay = ٢٥٠٠;              //milliseconds
١٤٣.        final Handler h = new Handler();
١٤٤.        h.postDelayed(new Runnable() {
١٤٥.            public void run() {
١٤٦.                phoneStateSetup();
١٤٧.
١٤٨.                if (signalstr != ٠) {
١٤٩.                    product product = new product(٠, String.valueOf(signalstr), St
    ring.valueOf(level), String.valueOf(CELLid), String.valueOf(LAC));
١٥٠.                    dbHandler.addproduct(product);
١٥١.                    sendingdata();
١٥٢.
١٥٣.                    final Handler handler = new Handler();
١٥٤.                    handler.postDelayed(new Runnable() {
١٥٥.                        @Override
١٥٦.                        public void run() {
١٥٧.                            // Do something after ١.٥s = ١٥٠٠٠ms
١٥٨.                            usb.setText(" ");
١٥٩.                        }
١٦٠.                    }, ١٥٠٠);
١٦١.                }
١٦٢.                Log.i("Tag", "timer repeated");
١٦٣.                h.postDelayed(this, delay);
١٦٤.            }
١٦٥.        }, delay);
١٦٦.
١٦٧.        start();
١٦٨.    }
١٦٩.
١٧٠.  public  void phoneStateSetup() {
١٧١.    try {
١٧٢.    Tm.listen(MPhoneStateListener, PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
١٧٣.    } catch (Exception e) {
١٧٤.        Toast.makeText(this,  e.toString(), Toast.LENGTH_LONG).show();
١٧٥.    }
١٧٦.
١٧٧.    MPhoneStateListener = new PhoneStateListener() {
١٧٨.
١٧٩.
١٨٠.        public void onSignalStrengthsChanged(SignalStrength signalStrength) {
١٨١.
١٨٢.
١٨٣.            signalstr = signalStrength.getGsmSignalStrength();
```

```
                signalstr = ((signalstr * ٢) - ١١٣);
                level = signalStrength.getLevel();

                SS.setText(String.valueOf(signalstr));
                LE.setText(String.valueOf(level));
                cellid.setText(String.valueOf(CELLid));
                lac.setText(String.valueOf(LAC));

            }
        };
}


    // start serial
    public void start ()
    {
        HashMap<String, UsbDevice> usbDevices = usbManager.getDeviceList();
        if (!usbDevices.isEmpty()) {
            boolean keep = true;
            for (Map.Entry<String, UsbDevice> entry : usbDevices.entrySet()) {
                device = entry.getValue();
                int deviceVID = device.getVendorId();
                if (deviceVID == ٠x٢٣٤١)//Arduino Vendor ID
                {
        PendingIntent pi = PendingIntent.getBroadcast(this, ٠, new Intent(ACTION_U
    SB_PERMISSION), ٠);
                    usbManager.requestPermission(device, pi);
                    keep = false;
                } else {
                    connection = null;
                    device = null;
                }

                if (!keep)
                    break;
            }
        }
    }

    private void tvAppend(TextView tv, CharSequence text)
    {
        final TextView ftv = tv;
        final CharSequence ftext = text;

        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                ftv.append(ftext);
            }
        });
    }

    public void sendingdata()
    {
        serialPort.write("a".getBytes());
        String send_٠ = String.valueOf(signalstr);
        serialPort.write(send_٠.getBytes());
        tvAppend(usb, "\nData Sent : " + "a:  " + send_٠ + "\n");

        serialPort.write("b".getBytes());
        String send_١ = String.valueOf(level);
        serialPort.write(send_١.getBytes());
        tvAppend(usb, "\nData Sent : " + "b:  " + send_١ + "\n");


```

```
٢٤٨.         serialPort.write("c".getBytes());
٢٤٩.         String send_٢ = String.valueOf(CELLid);
٢٥٠.         serialPort.write(send_٢.getBytes());
٢٥١.         tvAppend(usb, "\nData Sent : " + "c:  " + send_٢ + "\n");
٢٥٢.
٢٥٣.         serialPort.write("d".getBytes());
٢٥٤.         String send_٣ = String.valueOf(LAC);
٢٥٥.         serialPort.write(send_٣.getBytes());
٢٥٦.         tvAppend(usb, "\nData Sent : " + "d:  " + send_٣ + "\n");
٢٥٧.     }
٢٥٨.
٢٥٩.     @Override
٢٦٠.     public void onClick (View v)
٢٦١.     {
٢٦٢.         if (v == delete)
٢٦٣.         {
٢٦٤.             dbHandler.delete();
٢٦٥.
٢٦٦.         }
٢٦٧.
٢٦٨.         if (v == data)
٢٦٩.         {
٢٧٠.             Intent intent٣ = new Intent(this,DATABASE.class);
٢٧١.             startActivity(intent٣);
٢٧٢.         }
٢٧٣.     }
٢٧٤.
٢٧٥.}
```

General info activity code

```
١.  package com.example.bassem.final_project;
٢.
٣.  import android.annotation.TargetApi;
٤.  import android.app.PendingIntent;
٥.  import android.content.BroadcastReceiver;
٦.  import android.content.Context;
٧.  import android.content.Intent;
٨.  import android.content.IntentFilter;
٩.  import android.hardware.usb.UsbDevice;
١٠. import android.hardware.usb.UsbDeviceConnection;
١١. import android.hardware.usb.UsbManager;
١٢. import android.os.Build;
١٣. import android.os.Bundle;
١٤. import android.os.Handler;
١٥. import android.support.v٧.app.AppCompatActivity;
١٦. import android.telephony.TelephonyManager;
١٧. import android.telephony.gsm.GsmCellLocation;
١٨. import android.util.Log;
١٩. import android.view.View;
٢٠. import android.widget.Button;
٢١. import android.widget.TextView;
٢٢.
٢٣. import com.felhr.usbserial.UsbSerialDevice;
٢٤. import com.felhr.usbserial.UsbSerialInterface;
٢٥.
٢٦. import java.io.UnsupportedEncodingException;
٢٧. import java.util.HashMap;
٢٨. import java.util.Map;
٢٩.
٣٠. public class INFO extends AppCompatActivity  implements View.OnClickListener{
٣١.
٣٢.     TextView  operator , mcc , sim , rom , net , phone , imei , imsi , android , vo
    ice , usb;
٣٣.
٣٤.
٣٥.     int Mcc;
٣٦.     int SIM;
٣٧.     int NET;
٣٨.     int PHONE;
٣٩.     int counter  = ٠;
٤٠.     String SERIAL ;
٤١.     boolean roaming , voicecapable;
٤٢.     String operatorname , countrycode , IMEI , IMSI ;
٤٣.     Button send;
٤٤.
٤٥.     // usbserial
٤٦.
٤٧.     public final String ACTION_USB_PERMISSION = "com.hariharan.arduinousb.USB_PERMI
    SSION";
٤٨.     UsbManager usbManager;
٤٩.     UsbDevice device;
٥٠.     UsbSerialDevice serialPort;
٥١.     UsbDeviceConnection connection;
٥٢.
٥٣.     UsbSerialInterface.UsbReadCallback mCallback = new UsbSerialInterface.UsbReadCa
    llback()
٥٤.     {
٥٥.         @Override
٥٦.         public void onReceivedData(byte[] arg٠) {
٥٧.             String data = null;
٥٨.             try {
```

```
                data = new String(arg٠, "UTF-٨");
                data.concat("/n");
                tvAppend(usb, data);
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }


        }
    };

    private final BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            if (intent.getAction().equals(ACTION_USB_PERMISSION)) {
     boolean granted = intent.getExtras().getBoolean(UsbManager.EXTRA_PERMISSION_GR
ANTED);
                if (granted) {
                    connection = usbManager.openDevice(device);
         serialPort = UsbSerialDevice.createUsbSerialDevice(device, connection);
                    if (serialPort != null) {
                        if (serialPort.open()) {


                            serialPort.setBaudRate(٩٦٠٠);
                            serialPort.setDataBits(UsbSerialInterface.DATA_BITS_٨);
                            serialPort.setStopBits(UsbSerialInterface.STOP_BITS_١);
                            serialPort.setParity(UsbSerialInterface.PARITY_NONE);
                 serialPort.setFlowControl(UsbSerialInterface.FLOW_CONTROL_OFF);
                            serialPort.read(mCallback);
                            tvAppend(usb, "Serial Connection Opened!\n");
                        } else {
                            Log.d("SERIAL", "PORT NOT OPEN");
                        }
                    } else {
                        Log.d("SERIAL", "PORT IS NULL");
                    }
                } else {
                    Log.d("SERIAL", "PERM NOT GRANTED");
                }
   } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_ATTACHED)) {
                start();
  } else if (intent.getAction().equals(UsbManager.ACTION_USB_DEVICE_DETACHED)) {
                serialPort.close();
            }
        }
    };


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info);

        operator = (TextView) findViewById(R.id.Info_ON);
        mcc = (TextView) findViewById(R.id.Info_MCC);
        sim = (TextView) findViewById(R.id.Info_SIM);
        rom = (TextView) findViewById(R.id.Info_ROAMING);
        net = (TextView) findViewById(R.id.Info_NET);
        phone = (TextView) findViewById(R.id.Info_PHONE);
        imei = (TextView) findViewById(R.id.Info_IMEI);
        imsi = (TextView) findViewById(R.id.Info_IMSI);
        android = (TextView) findViewById(R.id.Info_ANDROID);
        voice = (TextView) findViewById(R.id.Info_VOICE);
        usb = (TextView)findViewById(R.id.Info_USB);
```

```
        send = (Button)findViewById(R.id.Info_SEND);

        send.setOnClickListener(this);

        usbManager = (UsbManager) getSystemService(this.USB_SERVICE);

        IntentFilter filter = new IntentFilter();
        filter.addAction(ACTION_USB_PERMISSION);
        filter.addAction(UsbManager.ACTION_USB_DEVICE_ATTACHED);
        filter.addAction(UsbManager.ACTION_USB_DEVICE_DETACHED);
        registerReceiver(broadcastReceiver, filter);

        calculations();
        start();

        if (counter == ٠)
        {
            final Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    // Do something after ٣.٥s = ٣٥٠٠ms
                    send.performClick();
                }
            }, ٣٥٠٠);
            counter =١;
        }
    }

    public void start() {

        HashMap<String, UsbDevice> usbDevices = usbManager.getDeviceList();
        if (!usbDevices.isEmpty()) {
            boolean keep = true;
            for (Map.Entry<String, UsbDevice> entry : usbDevices.entrySet()) {
                device = entry.getValue();
                int deviceVID = device.getVendorId();
                if (deviceVID == ٠x٢٣٤١)//Arduino Vendor ID
                {
                    PendingIntent pi = PendingIntent.getBroadcast(this, ٠, new Int
ent(ACTION_USB_PERMISSION), ٠);
                    usbManager.requestPermission(device, pi);
                    keep = false;
                } else {
                    connection = null;
                    device = null;
                }

                if (!keep)
                    break;
            }
        }
    }

    //tvappend

    private void tvAppend(TextView tv, CharSequence text) {
        final TextView ftv = tv;
        final CharSequence ftext = text;

        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                ftv.append(ftext);
```

```
١٨٦.             }
١٨٧.         });
١٨٨.     }
١٨٩.
١٩٠.     public void calculations()
١٩١.     {
١٩٢.
١٩٣.         TelephonyManager telephonyManager = (TelephonyManager) getSystemService(Co
    ntext.TELEPHONY_SERVICE);
١٩٤.
     GsmCellLocation cellLocation = (GsmCellLocation) telephonyManager.getCellLocation(
    );
١٩٥.
١٩٦.         // Operator name
١٩٧.
١٩٨.         operatorname = telephonyManager.getNetworkOperatorName();
١٩٩.         operator.setText(operatorname);
٢٠٠.
٢٠١.         // MCC
٢٠٢.
٢٠٣.         countrycode = telephonyManager.getNetworkOperator();
٢٠٤.         Mcc = Integer.parseInt(countrycode.substring(٠, ٣));
٢٠٥.         if (Mcc == ٦٠٢)
٢٠٦.         {
٢٠٧.             mcc.setText("Egypt (٦٠٢)");
٢٠٨.         }
٢٠٩.
٢١٠.
٢١١.         // SIM state
٢١٢.
٢١٣.         SIM = telephonyManager.getSimState();
٢١٤.         if (SIM == ٠)
٢١٥.         {
٢١٦.             sim.setText("UNKNOWN");
٢١٧.         }
٢١٨.         if (SIM == ١)
٢١٩.         {
٢٢٠.             sim.setText("NO SIM CARD");
٢٢١.         }
٢٢٢.         if (SIM == ٢)
٢٢٣.         {
٢٢٤.             sim.setText("PIN REQUIRED");
٢٢٥.         }
٢٢٦.         if (SIM == ٣)
٢٢٧.         {
٢٢٨.             sim.setText("PUK REQUIRED");
٢٢٩.         }
٢٣٠.         if (SIM == ٤)
٢٣١.         {
٢٣٢.             sim.setText("NETWORK LOCKED");
٢٣٣.         }
٢٣٤.         if (SIM == ٥)
٢٣٥.         {
٢٣٦.             sim.setText("Ready");
٢٣٧.         }
٢٣٨.
٢٣٩.
٢٤٠.         // in Roaming
٢٤١.
٢٤٢.         roaming = telephonyManager.isNetworkRoaming();
٢٤٣.         if (roaming == true)
٢٤٤.         {
٢٤٥.             rom.setText("YES");
٢٤٦.         }
٢٤٧.         else if (roaming == false)
```

```
٢٤٨.        {
٢٤٩.            rom.setText("NO");
٢٥٠.        }
٢٥١.
٢٥٢.        // Network Type
٢٥٣.
٢٥٤.        NET = telephonyManager.getNetworkType();
٢٥٥.        if (NET == ٠)
٢٥٦.        {
٢٥٧.            net.setText("UNKNOWN");
٢٥٨.        }
٢٥٩.        if (NET == ١)
٢٦٠.        {
٢٦١.            net.setText("GPRS");
٢٦٢.        }
٢٦٣.        if (NET == ٢)
٢٦٤.        {
٢٦٥.            net.setText("EDGE");
٢٦٦.        }
٢٦٧.        if (NET == ٣)
٢٦٨.        {
٢٦٩.            net.setText("UMTS");
٢٧٠.        }
٢٧١.        if (NET == ٤)
٢٧٢.        {
٢٧٣.            net.setText("CDMA");
٢٧٤.        }
٢٧٥.        if (NET == ٥)
٢٧٦.        {
٢٧٧.            net.setText("EVDO revision ٠");
٢٧٨.        }
٢٧٩.        if (NET == ٦)
٢٨٠.        {
٢٨١.            net.setText("EVDO revision A");
٢٨٢.        }
٢٨٣.        if (NET == ٧)
٢٨٤.        {
٢٨٥.            net.setText("١xRTT");
٢٨٦.        }
٢٨٧.        if (NET == ٨)
٢٨٨.        {
٢٨٩.            net.setText("HSDPA");
٢٩٠.        }
٢٩١.        if (NET == ٩)
٢٩٢.        {
٢٩٣.            net.setText("HSUPA");
٢٩٤.        }
٢٩٥.        if (NET == ١٠)
٢٩٦.        {
٢٩٧.            net.setText("HSPA");
٢٩٨.        }
٢٩٩.        if (NET == ١١)
٣٠٠.        {
٣٠١.            net.setText("IDEN");
٣٠٢.        }
٣٠٣.        if (NET == ١٢)
٣٠٤.        {
٣٠٥.            net.setText("EVDO revision B");
٣٠٦.        }
٣٠٧.        if (NET == ١٣)
٣٠٨.        {
٣٠٩.            net.setText("LTE ");
٣١٠.        }
٣١١.        if (NET == ١٤)
٣١٢.        {
```

```
٣١٣.            net.setText("EHRPD");
٣١٤.        }
٣١٥.        if (NET == ١٥)
٣١٦.        {
٣١٧.            net.setText("HSPAP");
٣١٨.        }
٣١٩.
٣٢٠.
٣٢١.        // Phone Type
٣٢٢.
٣٢٣.        PHONE = telephonyManager.getPhoneType();
٣٢٤.        if (PHONE == ٠)
٣٢٥.        {
٣٢٦.            phone.setText("NONE");
٣٢٧.
٣٢٨.        }
٣٢٩.        else if (PHONE == ١)
٣٣٠.        {
٣٣١.            phone.setText("GSM");
٣٣٢.
٣٣٣.        }
٣٣٤.        else if (PHONE == ٢)
٣٣٥.        {
٣٣٦.            phone.setText("CDMA");
٣٣٧.
٣٣٨.        }
٣٣٩.        else if (PHONE == ٣)
٣٤٠.        {
٣٤١.            phone.setText("SIP");
٣٤٢.
٣٤٣.        }
٣٤٤.
٣٤٥.        // Device ID (IMEI)
٣٤٦.
٣٤٧.        IMEI = telephonyManager.getDeviceId();
٣٤٨.        imei.setText(IMEI);
٣٤٩.
٣٥٠.        // Sim Serial Number (IMSI)
٣٥١.
٣٥٢.        IMSI = telephonyManager.getSubscriberId();
٣٥٣.        imsi.setText(IMSI);
٣٥٤.
٣٥٥.
٣٥٦.        // sim serial number
٣٥٧.
٣٥٨.        SERIAL = telephonyManager.getSimSerialNumber();
٣٥٩.        android.setText(SERIAL);
٣٦٠.
٣٦١.        // voice copable (circuit-switched)
٣٦٢.
٣٦٣.        voicecapable =telephonyManager.isVoiceCapable();
٣٦٤.        if (voicecapable == true)
٣٦٥.        {
٣٦٦.            voice.setText("YES");
٣٦٧.        }
٣٦٨.
٣٦٩.        else if (voicecapable == false)
٣٧٠.        {
٣٧١.            voice.setText("DATA ONLY");
٣٧٢.        }
٣٧٣.    }
٣٧٤.
٣٧٥.    public void sendingdata ()
٣٧٦.    {
٣٧٧.
```


```
٣٧٨.          serialPort.write("%".getBytes());
٣٧٩.          String send_٢ = operatorname;
٣٨٠.          serialPort.write(send_٢.getBytes());
٣٨١.
٣٨٢.          serialPort.write("f".getBytes());
٣٨٣.          String send_٣ = String.valueOf(Mcc);
٣٨٤.          serialPort.write(send_٣.getBytes());
٣٨٥.
٣٨٦.          serialPort.write("g".getBytes());
٣٨٧.          String send_٤ = String.valueOf(SIM);
٣٨٨.          serialPort.write(send_٤.getBytes());
٣٨٩.
٣٩٠.          serialPort.write("i".getBytes());
٣٩١.          if (roaming = true)
٣٩٢.          {
٣٩٣.              String send_٥ = "٠";
٣٩٤.              serialPort.write(send_٥.getBytes());
٣٩٥.          }
٣٩٦.          else if (roaming = false)
٣٩٧.          {
٣٩٨.              String send_٦ = "١";
٣٩٩.              serialPort.write(send_٦.getBytes());
٤٠٠.          }
٤٠١.
٤٠٢.          serialPort.write("h".getBytes());
٤٠٣.          String send_٧ = String.valueOf(NET);
٤٠٤.          serialPort.write(send_٧.getBytes());
٤٠٥.
٤٠٦.          serialPort.write("j".getBytes());
٤٠٧.          String send_٨ = String.valueOf(PHONE);
٤٠٨.          serialPort.write(send_٨.getBytes());
٤٠٩.
٤١٠.
٤١١.          serialPort.write("k".getBytes());
٤١٢.          String send_٩ = IMEI;
٤١٣.          serialPort.write(send_٩.getBytes());
٤١٤.
٤١٥.
٤١٦.          serialPort.write("l".getBytes());
٤١٧.          String send_١٠ = IMSI;
٤١٨.          serialPort.write(send_١٠.getBytes());
٤١٩.
٤٢٠.
٤٢١.          serialPort.write("m".getBytes());
٤٢٢.          if (voicecapable = true) {
٤٢٣.              String send_١١ = "١";
٤٢٤.              serialPort.write(send_١١.getBytes());
٤٢٥.
٤٢٦.          }
٤٢٧.          else if (voicecapable = false) {
٤٢٨.              String send_١٢ = "٠";
٤٢٩.              serialPort.write(send_١٢.getBytes());
٤٣٠.
٤٣١.          }
٤٣٢.
٤٣٣.          serialPort.write("n".getBytes());
٤٣٤.          String send_١٣ = SERIAL;
٤٣٥.          serialPort.write(send_١٣.getBytes());
٤٣٦.
٤٣٧.      }
٤٣٨.
٤٣٩.
٤٤٠.      @Override
٤٤١.      public void onClick(View v)
٤٤٢.      {
```

```
        if (v == send)
        {
            sendingdata();
            serialPort.close();
            tvAppend(usb, "Serial Connection Closed!\n");
        }
    }
}
```

My database handler code

```
package com.example.bassem.final_project;


import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.util.ArrayList;


public class MyDBHandler extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "products.db";
    public static final String TABLE_PRODUCTS = "products";
    public static final String COLOUM_ID = "id";
    public static final String COLOUM_SIGNAL = "signal";
    public static final String COLOUM_LEVEL = "level";
    public static final String COLOUM_CELLID = "cellid";
    public static final String COLOUM_LAC = "lac";


    public MyDBHandler(Context context)
    {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }









  @Override
    public void onCreate(SQLiteDatabase db)
    {

        String query = "CREATE TABLE " + TABLE_PRODUCTS + "(" +

                COLOUM_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "+

                COLOUM_SIGNAL + " TEXT, " +
```

```
                COLOUM_CELLID + " TEXT, " +

                COLOUM_LAC + " TEXT, " +

                COLOUM_LEVEL + " TEXT )";
        db.execSQL(query);

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL("DROP TABLE IF EXIST" + TABLE_PRODUCTS) ;
        onCreate(db);

    }

    public void addproduct (product product)
    {
        Log.i("Tag", "add product");


        SQLiteDatabase db = getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COLOUM_SIGNAL , product.getName());
        contentValues.put(COLOUM_LEVEL ,product.getName١());
        contentValues.put(COLOUM_CELLID,product.getName٢());
        contentValues.put(COLOUM_LAC,product.getName٣());
        db.insert(TABLE_PRODUCTS, null, contentValues);
        db.close();
    }


    public void deleteproduct (String productname)
    {

        SQLiteDatabase db = getReadableDatabase();           // reference
        db.execSQL("DELETE FROM " + TABLE_PRODUCTS + " WHERE " +  COLOUM_SIGNAL + "
=\"" + productname + "\";");
    }


  public void delete ()
  {
      SQLiteDatabase db = this.getWritableDatabase(); //get database
      db.execSQL("DROP TABLE IF EXISTS " + TABLE_PRODUCTS);


String query = "CREATE TABLE " + TABLE_PRODUCTS + "(" +

                COLOUM_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "+

                COLOUM_SIGNAL + " TEXT, " +

                COLOUM_CELLID + " TEXT, " +

                COLOUM_LAC + " TEXT, " +

                COLOUM_LEVEL + " TEXT )";
        db.execSQL(query);
        }
```

```
١١٥.
١١٦.
١١٧.
١١٨.     public ArrayList<product> getdata()
١١٩.     {
١٢٠.         Log.i("Tag","get data");
١٢١.         ArrayList<product> snapshotList = new ArrayList<product>();
١٢٢.         // Select All Query
١٢٣.         String selectQuery = "SELECT  * FROM " + TABLE_PRODUCTS + " WHERE ١";
١٢٤.
١٢٥.         SQLiteDatabase db = this.getWritableDatabase();
١٢٦.         Cursor cursor = db.rawQuery(selectQuery, null);
١٢٧.
١٢٨.         // looping through all rows and adding to list
١٢٩.         if (cursor.moveToFirst()) {
١٣٠.             do
١٣١.             {
١٣٢.                 product snapshot = new product(cursor.getInt(٠),cursor.getString(١
    ),cursor.getString(٢) , cursor.getString(٣) , cursor.getString(٤));
١٣٣.
١٣٤.                 // Adding contact to list
١٣٥.                 snapshotList.add(snapshot);
١٣٦.             } while (cursor.moveToNext());
١٣٧.         }
١٣٨.         db.close();
١٣٩.
١٤٠.         // return contact list
١٤١.         return snapshotList;
١٤٢.
١٤٣.     }
١٤٤.}
```

Product code

```
1.  package com.example.bassem.final_project;
2.
3.
4.  public class product
5.  {
6.      private int id;
7.      private String name;
8.      private String name1;
9.      private String name2;
10.     private String name3;
11.
12.
13.
14.
15. public product(int id ,String name , String name1 , String name2 , String name3)
16.     {
17.
18.         this.id=id;
19.         this.name = name;
20.         this.name1 = name1;
21.         this.name2 = name2;
22.         this.name3 = name3;
23.     }
24.
25.
26.     public void setId(int id) {
27.         this.id = id;
28.     }
29.
30.     public  int getId() {return id;}
31.
32.     public void setName(String name) {
33.         this.name = name;
34.     }
35.
36.     public String getName() {
37.         return name;
38.     }
39.
40.     public String getName1() {
41.         return name1;
42.     }
43.
44.     public void setName1(String name1) {
45.         this.name1 = name1;
46.     }
47.
48.     public String getName2() {
49.         return name2;
50.     }
51.
52.     public void setName2(String name2) {
53.         this.name2 = name2;
54.     }
55.
56.     public String getName3() {
57.         return name3;
58.     }
59.
60.     public void setName3(String name3) {
61.         this.name3 = name3;
62.     }
63. }
```

Customer adapter activity code

```
١.  package com.example.bassem.final_project;
٢.
٣.  import android.content.Context;
٤.  import android.view.LayoutInflater;
٥.  import android.view.View;
٦.  import android.view.ViewGroup;
٧.  import android.widget.ArrayAdapter;
٨.  import android.widget.ImageView;
٩.  import android.widget.TextView;
١٠.
١١. import java.util.ArrayList;
١٢.
١٣. class CustomAdapter extends ArrayAdapter<product>
١٤. {
١٥.
١٦.
١٧.     public CustomAdapter(Context context,ArrayList<product> datalist) {
١٨.         super(context,R.layout.custom_row,datalist);
١٩.     }
٢٠.
٢١.     @Override
٢٢.     public View getView(int position, View convertView, ViewGroup parent) {
٢٣.
٢٤.
٢٥.         LayoutInflater myinfaltor = LayoutInflater.from(getContext());
٢٦.         View customview = myinfaltor.inflate(R.layout.custom_row, parent, false);
٢٧.
٢٨.         product product = getItem(position);
٢٩.
٣٠.
٣١.         TextView text = (TextView) customview.findViewById(R.id.MAIN_TEXT);
٣٢.         TextView text١ = (TextView) customview.findViewById(R.id.MAIN_TEXT١);
٣٣.         TextView text٢ = (TextView) customview.findViewById(R.id.MAIN_TEXT٢);
٣٤.         TextView text٣ = (TextView) customview.findViewById(R.id.MAIN_TEXT٣);
٣٥.         TextView text٤ = (TextView) customview.findViewById(R.id.MAIN_TEXT٤);
٣٦.
٣٧.
٣٨.         text.setText(String.valueOf(product.getId()));
٣٩.         text١.setText(product.getName());
٤٠.         text٢.setText(product.getName١());
٤١.         text٣.setText(product.getName٢());
٤٢.         text٤.setText(product.getName٣());
٤٣.         return  customview;
٤٤.     }
٤٥.
٤٦.
٤٧. }
```

Database activity code

```
١. package com.example.bassem.final_project;
٢.
٣. import android.app.Activity;
٤. import android.os.Bundle;
٥. import android.os.Handler;
٦. import android.support.design.widget.FloatingActionButton;
٧. import android.support.design.widget.Snackbar;
٨. import android.support.v٧.app.AppCompatActivity;
٩. import android.support.v٧.widget.Toolbar;
١٠. import android.telephony.PhoneStateListener;
١١. import android.telephony.SignalStrength;
١٢. import android.telephony.TelephonyManager;
١٣. import android.util.Log;
١٤. import android.view.View;
١٥. import android.widget.AdapterView;
١٦. import android.widget.ListAdapter;
١٧. import android.widget.ListView;
١٨. import android.widget.Toast;
١٩.
٢٠. import java.util.ArrayList;
٢١.
٢٢. public class DATABASE extends AppCompatActivity {
٢٣.
٢٤.     ListView list;
٢٥.     ArrayList<product> datalist;
٢٦.     MyDBHandler dbHandler;
٢٧.
٢٨.     @Override
٢٩.     protected void onCreate(Bundle savedInstanceState) {
٣٠.         super.onCreate(savedInstanceState);
٣١.         setContentView(R.layout.activity_database);
٣٢.         Log.i("Tag", "database activity created");
٣٣.
٣٤.         final Handler h = new Handler();
٣٥.         final int delay = ٢٥٠٠;                              //milliseconds
٣٦.
٣٧.         h.postDelayed(new Runnable() {
٣٨.             public void run() {
٣٩.                 Log.i("Tag", "database timer");
٤٠.                 populatelistview();
٤١.                 h.postDelayed(this, delay);
٤٢.             }
٤٣.         }, delay);
٤٤.     }
٤٥.
٤٦.
٤٧.     private void populatelistview()
٤٨.     {
٤٩.         dbHandler = new MyDBHandler(this);
٥٠.         datalist = dbHandler.getdata();
٥١.         ListAdapter myadapter = new CustomAdapter (this,datalist);
٥٢.
٥٣.         // configure the list view
٥٤.         list = (ListView)findViewById(R.id.Main_list);
٥٥.         list.setAdapter(myadapter);
٥٦.
٥٧.         // on item click listner
٥٨.         list.setOnItemClickListener(
٥٩.                 new AdapterView.OnItemClickListener() {
٦٠.                     @Override
٦١. public void onItemClick(AdapterView<?> parent, View view, int position, long id
```

```
٦٢.                         {
٦٣.                             String ID = String.valueOf(id);
٦٤.                          Toast.makeText(DATABASE.this,ID,Toast.LENGTH_LONG).show();
٦٥.                         }
٦٦.                     }
٦٧.         );
٦٨.     }
٦٩. }
```

Android manifest code

```
<?xml version="١.٠" encoding="utf-٨"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bassem.final_project">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-feature android:name="android.hardware.usb.host" />
    <uses-sdk android:minSdkVersion="١٢" />


    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_info_black_٢٤dp"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".GSM"
            android:label="@string/title_activity_gsm"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".CDMA"
            android:label="@string/title_activity_cdm"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".INFO"
            android:label="@string/title_activity_info"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".DATABASE"
            android:label="DATABASE"
            android:theme="@style/AppTheme.NoActionBar"></activity>
    </application>

</manifest>
```

Resources xml device_filter

```
<resources>
    <usb-device vendor-id="٩٠٢٥" />
    <!-- Vendor ID of Arduino -->
</resources>
```

```
Library used in usb host (otg)

Usbserial.jar
```

Arduino Nano Code

```
#include <Wire.h>

float pid_p_gain_roll = 1.4;
float pid_i_gain_roll = 0.05;
float pid_d_gain_roll = 15;
int pid_max_roll = 400;

float pid_p_gain_pitch = pid_p_gain_roll;
float pid_i_gain_pitch = pid_i_gain_roll;
float pid_d_gain_pitch = pid_d_gain_roll;
int pid_max_pitch = pid_max_roll;

float pid_p_gain_yaw = 4.0;
float pid_i_gain_yaw = 0.02;
float pid_d_gain_yaw = 0.0;
int pid_max_yaw = 400;


byte last_channel_1, last_channel_2, last_channel_3, last_channel_4, last_channel_5, last_channel_6;
byte highByte, lowByte;
int receiver_input_channel_1, receiver_input_channel_2, receiver_input_channel_3, receiver_input_channel_4;
int receiver_input_channel_5, receiver_input_channel_6;
int receiver_channel_3_last;
int landing_throttle;
byte landing, landing_counter;
int counter_channel_1, counter_channel_2, counter_channel_3, counter_channel_4, loop_counter;
int esc_1, esc_2, esc_3, esc_4;
int throttle, battery_voltage;
int cal_int, start, gyro_address;
unsigned long timer_channel_1, timer_channel_2, timer_channel_3, timer_channel_4, esc_timer, esc_loop_timer;
unsigned long timer_1, timer_2, timer_3, timer_4, timer_5, timer_6, current_time;

unsigned long loop_timer;
double gyro_pitch, gyro_roll, gyro_yaw;
double gyro_axis[4], gyro_axis_cal[4];
float pid_error_temp;
float pid_i_mem_roll, pid_roll_setpoint, gyro_roll_input, pid_output_roll, pid_last_roll_d_error;
float pid_i_mem_pitch, pid_pitch_setpoint, gyro_pitch_input, pid_output_pitch, pid_last_pitch_d_error;
float pid_i_mem_yaw, pid_yaw_setpoint, gyro_yaw_input, pid_output_yaw, pid_last_yaw_d_error;


void setup(){

  Wire.begin();

  DDRD |= B11110000;
  DDRB |= B00000000;
  DDRC |= B00000000;

  receiver_channel_5_last = 1000;
  receiver_channel_3_last = 1000;

```

```
٥٤.   delay(1222);
٥٥.
٥٦.   Wire.beginTransmission(105);
٥٧.   Wire.write(0x20);
٥٨.   Wire.write(0x0F);
٥٩.   Wire.endTransmission();
٦٠.
٦١.   Wire.beginTransmission(105);
٦٢.   Wire.write(0x23);
٦٣.   Wire.write(0x90);
٦٤.   Wire.endTransmission();
٦٥.
٦٦.   delay(250);
٦٧.
٦٨.   for (cal_int = 0; cal_int < 2000 ; cal_int ++){
٦٩.     if(cal_int % 15 == 0)digitalWrite(12, !digitalRead(12));
٧٠.     gyro_signalen();
٧١.     gyro_roll_cal += gyro_roll;
٧٢.     gyro_pitch_cal += gyro_pitch;
٧٣.     gyro_yaw_cal += gyro_yaw;
٧٤.     PORTD |= B11110000;
٧٥.     delayMicroseconds(1000);
٧٦.     PORTD &= B00001111;
٧٧.     delayMicroseconds(1222);
٧٨.   }
٧٩.   gyro_roll_cal /= 2000;
٨٠.   gyro_pitch_cal /= 2000;
٨١.   gyro_yaw_cal /= 2000;
٨٢.
٨٣.    PCICR |= (1 << PCIE0);
٨٤.   PCICR |= (1 << PCIE1);
٨٥.   PCICR |= (1 << PCIE2);
٨٦.
٨٧.   PCMSK0 |= (1 << PCINT0);
٨٨.   PCMSK0 |= (1 << PCINT1);
٨٩.   PCMSK0 |= (1 << PCINT2);
٩٠.   PCMSK0 |= (1 << PCINT3);
٩١.
٩٢.   PCMSK1 |= (1 << PCINT8);
٩٣.   PCMSK1 |= (1 << PCINT9);
٩٤.   PCMSK1 |= (1 << PCINT10);
٩٥.   PCMSK1 |= (1 << PCINT11);
٩٦.
٩٧.   PCMSK0 |= (1 << PCINT4);
٩٨.   PCMSK2 |= (1 << PCINT18);
٩٩.
١٠٠.
١٠١.         while(receiver_input_channel_3 < 990 || receiver_input_channel_3 > 1020 |
    | receiver_input_channel_4 < 1400){
١٠٢.           start ++;
١٠٣.
١٠٤.           PORTD |= B11110000;
١٠٥.           delayMicroseconds(1000);
١٠٦.           PORTD &= B00001111;
١٠٧.           delayMicroseconds(1000);
١٠٨.           if(start == 125){
١٠٩.             start = 0;
١١٠.           }
١١١.         }
١١٢.         start = 0;
١١٣.
١١٤.         battery_voltage = (analogRead(7) + 65) * 1.2317;
١١٥.           }
١١٦.
١١٧.       void loop(){
```

```
١١٨.
١١٩.        if(landing == 1)
١٢٠.          {
١٢١.            emergency_landing();
١٢٢.          }
١٢٣.
١٢٤.          gyro_signalen();
١٢٥.          gyro_roll_input = (gyro_roll_input * 0.8) + ((gyro_roll / 57.14286) * 0.2
    );
١٢٦.          gyro_pitch_input = (gyro_pitch_input * 0.8) + ((gyro_pitch / 57.14286) *
    0.2);
١٢٧.          gyro_yaw_input = (gyro_yaw_input * 0.8) + ((gyro_yaw / 57.14286) * 0.2);

١٢٨.
١٢٩.
١٣٠.          if(receiver_input_channel_3 < 1050 && receiver_input_channel_4 < 1050)sta
    rt = 1;
١٣١.
١٣٢.          if(start == 1 && receiver_input_channel_3 < 1050 && receiver_input_channe
    l_4 > 1450){
١٣٣.            start = 2;
١٣٤.
١٣٥.            pid_i_mem_roll = 0;
١٣٦.            pid_last_roll_d_error = 0;
١٣٧.            pid_i_mem_pitch = 0;
١٣٨.            pid_last_pitch_d_error = 0;
١٣٩.            pid_i_mem_yaw = 0;
١٤٠.            pid_last_yaw_d_error = 0;
١٤١.          }
١٤٢.
١٤٣.          if(start == 2 && receiver_input_channel_3 < 1050 && receiver_input_channe
    l_4 > 1950)start = 0;
١٤٤.
١٤٥.
١٤٦.          pid_roll_setpoint = 0;
١٤٧.
١٤٨.          if(receiver_input_channel_1 > 1508)pid_roll_setpoint = (receiver_input_ch
    annel_1 - 1508)/3.0;
١٤٩.          else if(receiver_input_channel_1 < 1492)pid_roll_setpoint = (receiver_inp
    ut_channel_1 - 1492)/3.0;
١٥٠.
١٥١.
١٥٢.          pid_pitch_setpoint = 0;
١٥٣.
١٥٤.          if(receiver_input_channel_2 > 1508)pid_pitch_setpoint = (receiver_input_c
    hannel_2 - 1508)/3.0;
١٥٥.          else if(receiver_input_channel_2 < 1492)pid_pitch_setpoint = (receiver_in
    put_channel_2 - 1492)/3.0;
١٥٦.
١٥٧.
١٥٨.          pid_yaw_setpoint = 0;
١٥٩.
١٦٠.          if(receiver_input_channel_3 > 1050){
١٦١.            if(receiver_input_channel_4 > 1508)pid_yaw_setpoint = (receiver_input_c
    hannel_4 - 1508)/3.0;
١٦٢.            else if(receiver_input_channel_4 < 1492)pid_yaw_setpoint = (receiver_in
    put_channel_4 - 1492)/3.0;
١٦٣.          }
١٦٤.
١٦٥.          calculate_pid();
١٦٦.
١٦٧.
١٦٨.          battery_voltage = battery_voltage * 0.92 + (analogRead(7) + 65) * 0.09853
    ;
١٦٩.
```

```
١٧٠.          throttle = receiver_input_channel_3;
١٧١.
١٧٢.          if (start == 2){
١٧٣.            if (throttle > 1800) throttle = 1800;
١٧٤.            esc_1 = throttle - pid_output_pitch + pid_output_roll -
      pid_output_yaw;
١٧٥.            esc_2 = throttle + pid_output_pitch + pid_output_roll + pid_output_yaw;
١٧٦.            esc_3 = throttle + pid_output_pitch - pid_output_roll -
      pid_output_yaw;
١٧٧.            esc_4 = throttle - pid_output_pitch -
      pid_output_roll + pid_output_yaw;
١٧٨.
١٧٩.            if (battery_voltage < 1240 && battery_voltage > 800){
١٨٠.              esc_1 += esc_1 * ((1240 -
      battery_voltage)/(float)3500);
١٨١.              esc_2 += esc_2 * ((1240 -
      battery_voltage)/(float)3500);
١٨٢.              esc_3 += esc_3 * ((1240 -
      battery_voltage)/(float)3500);
١٨٣.              esc_4 += esc_4 * ((1240 -
      battery_voltage)/(float)3500);
١٨٤.            }
١٨٥.
١٨٦.            if (esc_1 < 1200) esc_1 = 1200;
١٨٧.            if (esc_2 < 1200) esc_2 = 1200;
١٨٨.            if (esc_3 < 1200) esc_3 = 1200;
١٨٩.            if (esc_4 < 1200) esc_4 = 1200;
١٩٠.
١٩١.            if(esc_1 > 2000)esc_1 = 2000;
١٩٢.            if(esc_2 > 2000)esc_2 = 2000;
١٩٣.            if(esc_3 > 2000)esc_3 = 2000;
١٩٤.            if(esc_4 > 2000)esc_4 = 2000;
١٩٥.          }
١٩٦.
١٩٧.          else{
١٩٨.            esc_1 = 1000;
١٩٩.            esc_2 = 1000;
٢٠٠.            esc_3 = 1000;
٢٠١.            esc_4 = 1000;
٢٠٢.          }
٢٠٣.
٢٠٤.
٢٠٥.          while(micros() -
      loop_timer < 2222);
٢٠٦.          loop_timer = micros();
٢٠٧.          timer_channel_1 = esc_1 + loop_timer;
```

```
٢٠٨.          timer_channel_2 = esc_2 + loop_timer;
٢٠٩.          timer_channel_3 = esc_3 + loop_timer;
٢١٠.          timer_channel_4 = esc_4 + loop_timer;
٢١١.
٢١٢.          while(PORTD >= 16){
٢١٣.            esc_loop_timer = micros();
٢١٤.            if(timer_channel_1 <= esc_loop_timer)PORTD &= B11101111;
٢١٥.            if(timer_channel_2 <= esc_loop_timer)PORTD &= B11011111;
٢١٦.            if(timer_channel_3 <= esc_loop_timer)PORTD &= B10111111;
٢١٧.            if(timer_channel_4 <= esc_loop_timer)PORTD &= B01111111;
٢١٨.          }
٢١٩.        }
٢٢٠.
٢٢١.
٢٢٢.
٢٢٣.        ISR(PCINT0_vect){
٢٢٤.          current_time = micros();
٢٢٥.
٢٢٦.
٢٢٧.
٢٢٨.          if(PINB & B00010000 ){
٢٢٩.            if(last_channel_5 == 0){
٢٣٠.              last_channel_5 = 1;
٢٣١.              timer_5 = current_time;
٢٣٢.            }
٢٣٣.          }
٢٣٤.          else if(last_channel_5 == 1){
٢٣٥.            last_channel_5 = 0;
٢٣٦.            receiver_input_channel_5 = current_time - timer_5;
٢٣٧.          }
٢٣٨.
٢٣٩.
٢٤٠.
٢٤١.
٢٤٢.          if(landing == 0)
٢٤٣.          {
٢٤٤.            if(receiver_input_channel_5 > 1500)
٢٤٥.            {
٢٤٦.              if(PINB & B00000001){
٢٤٧.                if(last_channel_1 == 0){
٢٤٨.                  last_channel_1 = 1;
٢٤٩.                  timer_1 = current_time;
٢٥٠.                }
٢٥١.              }
٢٥٢.              else if(last_channel_1 == 1){
٢٥٣.                last_channel_1 = 0;
٢٥٤.                receiver_input_channel_1 = current_time -
    timer_1;
٢٥٥.              }
٢٥٦.              if(PINB & B00000010 ){
٢٥٧.                if(last_channel_2 == 0){
٢٥٨.                  last_channel_2 = 1;
٢٥٩.                  timer_2 = current_time;
٢٦٠.                }
٢٦١.              }
٢٦٢.              else if(last_channel_2 == 1){
```

```
                last_channel_2 = 0;
                receiver_input_channel_2 = current_time -
    timer_2;
              }
                if(PINB & B00000100 ){
                  if(last_channel_3 == 0){
                    last_channel_3 = 1;
                    timer_3 = current_time;
                  }
                }
                else if(last_channel_3 == 1){
                  last_channel_3 = 0;
                  receiver_input_channel_3 = current_time -
    timer_3;

                }
              if(PINB & B00001000 ){
                if(last_channel_4 == 0){
                  last_channel_4 = 1;
                  timer_4 = current_time;
                }
              }
              else if(last_channel_4 == 1){
                last_channel_4 = 0;
                receiver_input_channel_4 = current_time -
    timer_4;
              }
            }
          }
        }


        ISR(PCINT1_vect)
        {
          current_time = micros();
          if(landing == 0)
          {
             if(receiver_input_channel_5 < 1500)
            {
              if(PINC & B00000001){
                if(last_channel_1 == 0){
                  last_channel_1 = 1;
                  timer_1 = current_time;
                }
              }
              else if(last_channel_1 == 1){
                last_channel_1 = 0;
                receiver_input_channel_1 = current_time -
    timer_1;
              }
              if(PINC & B00000010 ){
                if(last_channel_2 == 0){
                  last_channel_2 = 1;
                  timer_2 = current_time;
                }
              }
              else if(last_channel_2 == 1){
                last_channel_2 = 0;
                receiver_input_channel_2 = current_time -
    timer_2;
              }
                if(PINC & B00000100 ){
                  if(last_channel_3 == 0){
                    last_channel_3 = 1;
                    timer_3 = current_time;
```

```
٣٢٣.                 }
٣٢٤.               }
٣٢٥.               else if(last_channel_3 == 1){
٣٢٦.                 last_channel_3 = 0;
٣٢٧.                 receiver_input_channel_3 = current_time -
     timer_3;
٣٢٨.
٣٢٩.               }
٣٣٠.
٣٣١.             if(PINC & B00001000 ){
٣٣٢.               if(last_channel_4 == 0){
٣٣٣.                 last_channel_4 = 1;
٣٣٤.                 timer_4 = current_time;
٣٣٥.               }
٣٣٦.             }
٣٣٧.             else if(last_channel_4 == 1){
٣٣٨.               last_channel_4 = 0;
٣٣٩.               receiver_input_channel_4 = current_time - timer_4;
٣٤٠.             }
٣٤١.           }
٣٤٢.         }
٣٤٣.       }
٣٤٤.
٣٤٥.
٣٤٦.
٣٤٧.       ISR(PCINT2_vect)
٣٤٨.       {
٣٤٩.
٣٥٠.
٣٥١.         current_time = micros();
٣٥٢.         if(PIND & B00000100 ){
٣٥٣.           if(last_channel_6 == 0){
٣٥٤.             last_channel_6 = 1;
٣٥٥.             timer_6 = current_time;
٣٥٦.           }
٣٥٧.         }
٣٥٨.         else if(last_channel_6 == 1){
٣٥٩.           last_channel_6 = 0;
٣٦٠.           receiver_input_channel_6 = current_time - timer_6;
٣٦١.
٣٦٢.           if(receiver_input_channel_6 > 1500)
٣٦٣.           {
٣٦٤.             if(landing == 0)
٣٦٥.             {
٣٦٦.               landing_throttle = receiver_input_channel_3;
٣٦٧.               landing = 1;
٣٦٨.             }
٣٦٩.
٣٧٠.           if(receiver_input_channel_6 < 1500)
٣٧١.           {
٣٧٢.             landing = 0;
٣٧٣.           }
٣٧٤.
٣٧٥.
٣٧٦.         }
٣٧٧.         }
٣٧٨.       }
٣٧٩.
٣٨٠.
٣٨١.
٣٨٢.
٣٨٣.
٣٨٤.
٣٨٥.       void gyro_signalen(){
٣٨٦.         Wire.beginTransmission(105);
```

```
٣٨٧.          Wire.write(168);
٣٨٨.          Wire.endTransmission();
٣٨٩.          Wire.requestFrom(105, 6);
٣٩٠.          while(Wire.available() < 6);
٣٩١.          lowByte = Wire.read();
٣٩٢.          highByte = Wire.read();
٣٩٣.          gyro_roll = ((highByte<<8)|lowByte);
٣٩٤.          if(cal_int == 2000)gyro_roll -= gyro_roll_cal;
٣٩٥.          lowByte = Wire.read();
٣٩٦.          highByte = Wire.read();
٣٩٧.          gyro_pitch = ((highByte<<8)|lowByte);
٣٩٨.          gyro_pitch *= -1;
٣٩٩.          if(cal_int == 2000)gyro_pitch -= gyro_pitch_cal;
٤٠٠.          lowByte = Wire.read();
٤٠١.          highByte = Wire.read();
٤٠٢.          gyro_yaw = ((highByte<<8)|lowByte);
٤٠٣.          gyro_yaw *= -1;
٤٠٤.          if(cal_int == 2000)gyro_yaw -= gyro_yaw_cal;
٤٠٥.        }
٤٠٦.
٤٠٧.
٤٠٨.
٤٠٩.        void calculate_pid(){
٤١٠.          pid_error_temp = gyro_roll_input - pid_roll_setpoint;
٤١١.          pid_i_mem_roll += pid_i_gain_roll * pid_error_temp;
٤١٢.          if(pid_i_mem_roll > pid_max_roll)pid_i_mem_roll = pid_max_roll;
٤١٣.          else if(pid_i_mem_roll < pid_max_roll * -
    1)pid_i_mem_roll = pid_max_roll * -1;
٤١٤.
٤١٥.          pid_output_roll = pid_p_gain_roll * pid_error_temp + pid_i_mem_roll + pid
    _d_gain_roll * (pid_error_temp - pid_last_roll_d_error);
٤١٦.          if(pid_output_roll > pid_max_roll)pid_output_roll = pid_max_roll;
٤١٧.          else if(pid_output_roll < pid_max_roll * -
    1)pid_output_roll = pid_max_roll * -1;
٤١٨.
٤١٩.          pid_last_roll_d_error = pid_error_temp;
٤٢٠.
٤٢١.          pid_error_temp = gyro_pitch_input - pid_pitch_setpoint;
٤٢٢.          pid_i_mem_pitch += pid_i_gain_pitch * pid_error_temp;
٤٢٣.          if(pid_i_mem_pitch > pid_max_pitch)pid_i_mem_pitch = pid_max_pitch;
٤٢٤.          else if(pid_i_mem_pitch < pid_max_pitch * -
    1)pid_i_mem_pitch = pid_max_pitch * -1;
٤٢٥.
٤٢٦.          pid_output_pitch = pid_p_gain_pitch * pid_error_temp + pid_i_mem_pitch +
    pid_d_gain_pitch * (pid_error_temp - pid_last_pitch_d_error);
٤٢٧.          if(pid_output_pitch > pid_max_pitch)pid_output_pitch = pid_max_pitch;
٤٢٨.          else if(pid_output_pitch < pid_max_pitch * -
    1)pid_output_pitch = pid_max_pitch * -1;
٤٢٩.
٤٣٠.          pid_last_pitch_d_error = pid_error_temp;
٤٣١.
٤٣٢.          pid_error_temp = gyro_yaw_input - pid_yaw_setpoint;
٤٣٣.          pid_i_mem_yaw += pid_i_gain_yaw * pid_error_temp;
٤٣٤.          if(pid_i_mem_yaw > pid_max_yaw)pid_i_mem_yaw = pid_max_yaw;
٤٣٥.          else if(pid_i_mem_yaw < pid_max_yaw * -1)pid_i_mem_yaw = pid_max_yaw * -
    1;
٤٣٦.
٤٣٧.          pid_output_yaw = pid_p_gain_yaw * pid_error_temp + pid_i_mem_yaw + pid_d_
    gain_yaw * (pid_error_temp - pid_last_yaw_d_error);
٤٣٨.          if(pid_output_yaw > pid_max_yaw)pid_output_yaw = pid_max_yaw;
٤٣٩.          else if(pid_output_yaw < pid_max_yaw * -
    1)pid_output_yaw = pid_max_yaw * -1;
٤٤٠.
٤٤١.          pid_last_yaw_d_error = pid_error_temp;
٤٤٢.        }
```

```
٤٤٣.        void emergency_landing()
٤٤٤.        {
٤٤٥.          if(landing == 1)
٤٤٦.          {
٤٤٧.            receiver_input_channel_1 = 1500;
٤٤٨.            receiver_input_channel_2 = 1500;
٤٤٩.            receiver_input_channel_4 = 1500;
٤٥٠.
٤٥١.            landing_counter++;
٤٥٢.
٤٥٣.            if(landing_throttle > 1400)
٤٥٤.            {
٤٥٥.              if(landing_counter >= 10)
٤٥٦.              {
٤٥٧.                landing_throttle--;
٤٥٨.                landing_counter = 0;
٤٥٩.              }
٤٦٠.            }
٤٦١.
٤٦٢.            if((landing_throttle <= 1400) && (landing_throttle >= 1250))
٤٦٣.            {
٤٦٤.               landing_throttle--;
٤٦٥.               landing_counter = 0;
٤٦٦.            }
٤٦٧.                  receiver_input_channel_3 = landing_throttle;
٤٦٨.
٤٦٩.            if(landing_throttle < 1250)
٤٧٠.            {
٤٧١.              start = 0;
٤٧٢.              landing = 0;
٤٧٣.              landing_counter = 0;
٤٧٤.            }
٤٧٥.          }
٤٧٦.        }
```

Ardunio Mega Code

```
١.
٢.  #include "TinyGPS++.h"
٣.  #include <Servo.h>
٤.  #include <NewPing.h>
٥.
٦.
٧.  //** GPS ublox neo 6M **//
٨.
٩.  TinyGPSPlus gps;
١٠.
١١. long satellite_number;
١٢. double latitude;
١٣. double longitude;
١٤. double copter_speed;
١٥. double copter_altitude;
١٦.
١٧. //** pwm generation **//
١٨. Servo roll;
١٩. Servo pitch;
٢٠. Servo throttle;
٢١. Servo yaw;
٢٢.
```

```
//** ultrasonic **//

NewPing ultra_sonic_1(8, 9, 500);

unsigned int us_ultra_sonic_1;
int distance_ultra_sonic_1;
long ping_time;
int time_ms_ping;


//** controller array **//
const int controller_length = 4;
byte controller_array_received[controller_length];
int controller_array[controller_length];
int controller_array_operator[controller_length];

//** gps arrays **//
float gps_receive_comp_array[2];

//** sending pulse variables**//
int sending[controller_length];

//** Serial1 1 variable **//
byte indentifier_char;

//** copter flight state **//
byte state;
// 0 --> copter is stopped
// 1 --> copter is working
// 2 --> copter is landed

//** error control variables **// 1.2
byte error;
byte communication_error;
long comm_error_start;

//** sendind data timimg **//
byte transmission;
long start_sending_time;
long time_now;
int sending_seconds;
byte sending_state;

//** mobile drivetest data **//
byte indicator_mobile;
int signal_strength;
int level;
unsigned int cell_id;
unsigned int location;
int operator_name;
//String operator_name_string;
int mcc;
int sim_state;
int roaming;
int network_type;
int phone_type;
byte imei[15];
byte imsi[15];
int voice_capable;
byte sim_serial[19];

byte general_info;

void setup() {

```

```
Serial.begin(9600);

Serial1.begin(9600);

Serial2.begin(9600);

controller_start();

for(int i = 0; i < controller_length; i++)
{
  if(i == 2)
  {
            controller_array[i] = 0;
            controller_array_received[i] = 0;
            controller_array_operator[i] = 0;
          }
          else
          {
            controller_array[i] = 127;
            controller_array_received[i] = 127;
            controller_array_operator[i] = 127;
          }
        }

        while (!Serial1) { ; }

        state = 0;
        communication_error = 0;
        start_sending_time = micros();
        sending_state = 0;
        transmission = 0;
        general_info = 30;
      }

      void loop() {

        //** communication error control **//

        //comm_error_control();

        //** receiving data from computer **//

        if (Serial1.available() > 0)
        {
          indentifier_char = Serial1.read();
          switch (indentifier_char)
          {

            case 'C':
              break;

            case 'r':
              controller_array[0] = Serial1.parseInt();
              if ((state == 1)||(state == 2))
              {
                controllers();
                state = 1;
                communication_error = 0;
                comm_error_start = 0;
              }
              break;

           case 'p':
              controller_array[1] = Serial1.parseInt();
              if ((state == 1)||(state == 2))
```

```
                {
                  controllers();
                  state = 1;
                  communication_error = 0;
                  comm_error_start = 0;
                }
                break;
              case 't':
                controller_array[2] = Serial1.parseInt();
                if ((state == 1)||(state == 2))
                {
                  controllers();
                  state = 1;
                  communication_error = 0;
                  comm_error_start = 0;
                }
                break;
              case 'y':
                controller_array[3] = Serial1.parseInt();
                if ((state == 1)||(state == 2))
                {
                  controllers();
                  state = 1;
                  communication_error = 0;
                  comm_error_start = 0;
                }
                break;


               case 'G':
                 gps_computer_receive();
                 communication_error = 0;
                 comm_error_start = 0;
                 break;

               case 'T':
                 start_sending_time = micros();
                 sending_state = 0;
                 transmission = 1;
                 communication_error = 0;
                 comm_error_start = 0;
                 break;

               case 'L':
                if (state == 1)
                {
                   landing();
                   state = 2;
                   communication_error = 0;
                   comm_error_start = 0;
                }
                 break;

               case 'S':
                 if((state == 0) || (controller_array[2] <= 50) )
                 {
                   start_copter();
                   state = 1;
                   communication_error = 0;
                   comm_error_start = 0;
                 }
                 break;

               case 'P':
                 if ((state == 2)||(controller_array[2] <= 50))
```


```
                {
                  stop_copter();
                  state = 0;
                  communication_error = 0;
                  comm_error_start = 0;
                }
                break;

              case 'N' :
                transmission = 0;
                communication_error = 0;
                comm_error_start = 0;
                break;

              default :
                error = indentifier_char;
                Serial1.write('E');
                Serial1.write(error);
                if (communication_error == 0)
                {
                  comm_error_start = micros();
                }
                communication_error = 1;
                break;

            }
          }

           if(Serial.available()> 0)
          {
            indicator_mobile = Serial.read();

            switch(indicator_mobile)
            {
            case 'a' :
              signal_strength = Serial.parseInt();
              break;
            case 'b' :
              level = Serial.parseInt();
              break;
            case 'c' :
              cell_id = Serial.parseInt();
              break;
            case 'd' :
              location = Serial.parseInt();
              break;
            case '%' :
              operator_name = Serial.parseInt();
              //Serial.readBytesUntil('z',operator_name);
              //readStringUntil('z',operator_name_string);
              break;
            case 'f':
              mcc = Serial.parseInt();
              break;
            case 'g' :
              sim_state = Serial.parseInt();
              break;
            case 'i' :
              roaming = Serial.parseInt();
              break;
            case 'h' :
              network_type = Serial.parseInt();
              break;
            case 'j' :
              phone_type =Serial.parseInt();
```

```
٢٨٣.              break;
٢٨٤.            case 'k' :
٢٨٥.              Serial.readBytes(imei, 15);
٢٨٦.              break;
٢٨٧.            case 'l' :
٢٨٨.              Serial.readBytes(imsi, 15);
٢٨٩.              break;
٢٩٠.            case 'm' :
٢٩١.              voice_capable = Serial.parseInt();
٢٩٢.              break;
٢٩٣.            case 'n' :
٢٩٤.              Serial.readBytes(sim_serial, 19);
٢٩٥.              break;
٢٩٦.            default:
٢٩٧.              break;
٢٩٨.            }
٢٩٩.          }
٣٠٠.
٣٠١.          //** GPS receiving data **//
٣٠٢.
٣٠٣.          while(Serial2.available())
٣٠٤.          {
٣٠٥.            gps.encode(Serial2.read());
٣٠٦.          }
٣٠٧.
٣٠٨.          if(gps.location.isUpdated())
٣٠٩.          {
٣١٠.
٣١١.            satellite_number = gps.satellites.value();
٣١٢.            latitude = gps.location.lat();
٣١٣.            longitude = gps.location.lng();
٣١٤.            copter_speed = gps.speed.mps();
٣١٥.            copter_altitude = gps.altitude.meters();
٣١٦.          }
٣١٧.
٣١٨.
٣١٩.          //** ultrasonic **//
٣٢٠.          //time_now_ping = micros();
٣٢١.          time_ms_ping = (int)((micros() - ping_time)/1000);
٣٢٢.          if(time_ms_ping > 29)
٣٢٣.          {
٣٢٤.            us_ultra_sonic_1 = ultra_sonic_1.ping();
٣٢٥.            distance_ultra_sonic_1 = (int)(us_ultra_sonic_1 / US_ROUNDTRIP_CM);
٣٢٦.          }
٣٢٧.
٣٢٨.
٣٢٩.          //** sendind data **//
٣٣٠.          if(transmission == 1)
٣٣١.          {
٣٣٢.             time_now = micros();
٣٣٣.             sending_seconds = (int)((time_now - start_sending_time)/500000);
٣٣٤.             if((sending_seconds >= 1)&&(sending_state == 0))
٣٣٥.             {
٣٣٦.                send_float(latitude, 'G');
٣٣٧.                send_float(longitude, 'D');
٣٣٨.
٣٣٩.                int copter_speed_int = (int)copter_speed;
٣٤٠.                Serial1.write('M');
٣٤١.                send_int_2digits(copter_speed_int);
٣٤٢.
٣٤٣.
٣٤٤.                Serial1.write('N');
٣٤٥.                send_int_2digits(satellite_number);
٣٤٦.
٣٤٧.                int copter_altitude_int = (int)copter_altitude;
```

```
٣٤٨.                Serial1.write('A');
٣٤٩.                send_int_2digits(copter_altitude_int);
٣٥٠.
٣٥١.                sending_state = 1;
٣٥٢.             }
٣٥٣.             if((sending_seconds >= 2)&&(sending_state == 1))
٣٥٤.             {
٣٥٥.                Serial1.write('H');
٣٥٦.                int battery1_health = analogRead(0);
٣٥٧.                byte x = map(battery1_health, 0, 1023, 0, 99);
٣٥٨.                send_int_2digits(x);
٣٥٩.
٣٦٠.                Serial1.write('B');
٣٦١.                int battery2_health = analogRead(1);
٣٦٢.                byte y = map(battery2_health, 1, 1023, 0, 99);
٣٦٣.                send_int_2digits(y);
٣٦٤.
٣٦٥.                Serial1.write('F');
٣٦٦.                int battery3_health = analogRead(2);
٣٦٧.                byte z = map(battery2_health, 1, 1023, 0, 99);
٣٦٨.                send_int_2digits(z);
٣٦٩.
٣٧٠.                sending_state = 3;
٣٧١.
٣٧٢.             }
٣٧٣.
٣٧٤.             if((sending_seconds >= 3)&&(sending_state == 2))
٣٧٥.             {
٣٧٦.                Serial1.write('a');
٣٧٧.                send_int_3digits(signal_strength);
٣٧٨.
٣٧٩.                Serial1.write('b');
٣٨٠.                Serial1.write((byte)level);
٣٨١.
٣٨٢.                general_info++;
٣٨٣.
٣٨٤.
٣٨٥.                sending_state = 0;
٣٨٦.                start_sending_time = micros();
٣٨٧.             }
٣٨٨.
٣٨٩.             if(general_info >= 40)
٣٩٠.             {
٣٩١.                Serial1.write('c');
٣٩٢.                send_int_5digits(cell_id);
٣٩٣.
٣٩٤.                Serial1.write('d');
٣٩٥.                send_int_5digits(location);
٣٩٦.
٣٩٧.                Serial1.write('o');
٣٩٨.                Serial1.write((byte)operator_name);
٣٩٩.
٤٠٠.                Serial1.write('f');
٤٠١.                send_int_3digits(mcc);
٤٠٢.
٤٠٣.                Serial1.write('g');
٤٠٤.                Serial1.write((byte)sim_state);
٤٠٥.
٤٠٦.                Serial1.write('i');
٤٠٧.                Serial1.write((byte)roaming);
٤٠٨.
٤٠٩.                Serial1.write('h');
٤١٠.                Serial1.write((byte)network_type);
٤١١.
٤١٢.                Serial1.write('j');
```

```
                Serial1.write((byte)phone_type);

                Serial1.write('k');
                Serial1.write(imei, 15);

                Serial1.write('l');
                Serial1.write(imsi, 15);

                Serial1.write('m');
                Serial1.write((byte)voice_capable);

                Serial1.write('n');
                Serial1.write(sim_serial, 19);

                general_info = 0;
             }
          }
        }


        void gps_computer_receive()
        {

          byte f;
          f = Serial1.read();
          if(f = 'f')
          {
            gps_receive_comp_array[0] = Serial1.parseFloat();
            f = Serial1.read();
            if(f = 'f')
            {
              gps_receive_comp_array[1] = Serial1.parseFloat();
            }
          }
        }

        void landing()
        {

          controller_array[0] = 127;
          controller_array[1] = 127;
          controller_array[3] = 127;

           if(controller_array[2] < 0)
           {
             controller_array[2] = 256 + controller_array[0];
           }

          while(controller_array[2] > 100 )
          {
            controller_array[2]--;
            controllers();

            if(controller_array[2] > 120)
            {
              delay(100);
            }
            else
            {
              delay(50);
            }
          }
          stop_copter();
        }

```

```
٤٧٨.
٤٧٩.
٤٨٠.        void send_float(float floating_point, byte indicator)
٤٨١.        {
٤٨٢.
٤٨٣.          Serial1.write(indicator);
٤٨٤.
٤٨٥.          int x = (int) floating_point;
٤٨٦.          long y;
٤٨٧.          long z;
٤٨٨.          float a = floating_point - x;
٤٨٩.          unsigned long w = (unsigned long)(a * 100000000);
٤٩٠.
٤٩١.          if((x > 99) && (x < 1000))
٤٩٢.          {
٤٩٣.             y = x / 100;
٤٩٤.             Serial1.write(y+48);
٤٩٥.             z = x % 100;
٤٩٦.             y = z / 10;
٤٩٧.             Serial1.write(y+48);
٤٩٨.             y = z % 10;
٤٩٩.             Serial1.write(y+48);
٥٠٠.          }
٥٠١.          if ((x > 9) && (x < 100))
٥٠٢.          {
٥٠٣.             y = x / 10;
٥٠٤.             Serial1.write(y+48);
٥٠٥.             y = x % 10;
٥٠٦.             Serial1.write(y+48);
٥٠٧.          }
٥٠٨.
٥٠٩.          if (x < 10)
٥١٠.          {
٥١١.            Serial1.write(x+48);
٥١٢.          }
٥١٣.
٥١٤.           Serial1.write('.');
٥١٥.           y = w / 10000000;
٥١٦.           Serial1.write(y+48);
٥١٧.           z = w % 10000000;
٥١٨.           y = z / 1000000;
٥١٩.           Serial1.write(y+48);
٥٢٠.           z = z % 1000000;
٥٢١.           y = z / 100000;
٥٢٢.           Serial1.write(y+48);
٥٢٣.           z = z % 100000;
٥٢٤.           y = z / 10000;
٥٢٥.           Serial1.write(y+48);
٥٢٦.           z = z % 10000;
٥٢٧.           y = z / 1000;
٥٢٨.           Serial1.write(y+48);
٥٢٩.           z = z % 1000;
٥٣٠.           y = z / 100;
٥٣١.           Serial1.write(y+48);
٥٣٢.        }
٥٣٣.
٥٣٤.        void pulse_generator()
٥٣٥.        {
٥٣٦.          for(int i = 0; i < controller_length; i++)
٥٣٧.          {
٥٣٨.           q = 3.92156863 *controller_array_operator[i];
٥٣٩.           sending[i] = (int)(q + 1000);
٥٤٠.          }
٥٤١.
٥٤٢.          roll.writeMicroseconds(sending[0]);
```

```
٥٤٣.          pitch.writeMicroseconds(sending[1]);
٥٤٤.          throttle.writeMicroseconds(sending[2]);
٥٤٥.          yaw.writeMicroseconds(sending[3]);
٥٤٦.        }
٥٤٧.
٥٤٨.        void start_copter()
٥٤٩.        {
٥٥٠.          controller_array[0]= 127;
٥٥١.          controller_array[1]= 127;
٥٥٢.          controller_array[2]= 0;
٥٥٣.          controller_array[3]= 127;
٥٥٤.          controllers();
٥٥٥.          delay(200);
٥٥٦.
٥٥٧.          controller_array[0]= 127;
٥٥٨.          controller_array[1]= 127;
٥٥٩.          controller_array[2]= 0;
٥٦٠.          controller_array[3]= 0;
٥٦١.          controllers();
٥٦٢.          delay(200);
٥٦٣.
٥٦٤.          controller_array[0]= 127;
٥٦٥.          controller_array[1]= 127;
٥٦٦.          controller_array[2]= 0;
٥٦٧.          controller_array[3]= 127;
٥٦٨.          controllers();
٥٦٩.          delay(50);
٥٧٠.        }
٥٧١.
٥٧٢.
٥٧٣.        void stop_copter()
٥٧٤.        {
٥٧٥.          controller_array[0]= 127;
٥٧٦.          controller_array[1]= 127;
٥٧٧.          controller_array[2]= 0;
٥٧٨.          controller_array[3]= 127;
٥٧٩.          controllers();
٥٨٠.          delay(200);
٥٨١.
٥٨٢.          controller_array[0]= 127;
٥٨٣.          controller_array[1]= 127;
٥٨٤.          controller_array[2]= 0;
٥٨٥.          controller_array[3]= 255;
٥٨٦.          controllers();
٥٨٧.          delay(200);
٥٨٨.
٥٨٩.          controller_array[0]= 127;
٥٩٠.          controller_array[1]= 127;
٥٩١.          controller_array[2]= 0;
٥٩٢.          controller_array[3]= 127;
٥٩٣.          controllers();
٥٩٤.          delay(50);
٥٩٥.        }
٥٩٦.
٥٩٧.        void send_int_2digits(byte x)
٥٩٨.        {
٥٩٩.          byte y;
٦٠٠.          y = x / 10;
٦٠١.          Serial1.write(y+48);
٦٠٢.          y = x % 10;
٦٠٣.          Serial1.write(y+48);
٦٠٤.        }
٦٠٥.
٦٠٦.        void send_int_3digits(int x)
٦٠٧.        {
```

```
  int y;
  y = x / 100;
  Serial1.write(y+48);
  int w = x % 100;
  y = w /10;
  Serial1.write(y+48);
  y = w % 10;
  Serial1.write(y+48);
}

void send_int_5digits(int z)
{
  int y ;
   y = z / 10000;
   Serial1.write(y+48);
   z = z % 10000;
   y = z / 1000;
   Serial1.write(y+48);
   z = z % 1000;
   y = z / 100;
   Serial1.write(y+48);
   z = z % 100;
   y = z / 10;
   Serial1.write(y+48);
   y = z % 10;
   Serial1.write(y+48);
}

void comm_error_control()
{
  long comm_error_period;
  long temp;
  long error_seconds;
  if((communication_error == 1)&&(state != 0))
  {
    temp = micros();
    comm_error_period = temp - comm_error_start;
    error_seconds = (long)(comm_error_period / 1000000);

    if((error_seconds >= 2)&&(error_seconds < 10))
    {
      controller_array[1]= 127;
      controller_array[2]= 127;
      controller_array[3]= 127;
      controllers();
    }
    if(error_seconds >= 10)
    {
      landing();
      state = 2;
    }
  }
}

void controllers()
{
  while((controller_array_operator[0] != controller_array[0])\
      ||(controller_array_operator[1] != controller_array[1])\
      ||(controller_array_operator[2] != controller_array[2])\
      ||(controller_array_operator[3] != controller_array[3]))
  {
    for(int i = 0; i < controller_length; i++)
    {
      if(controller_array_operator[i] < controller_array[i])
      {
```

```
٦٧٣.                controller_array_operator[i]++;
٦٧٤.              }
٦٧٥.              if(controller_array_operator[i] > controller_array[i])
٦٧٦.              {
٦٧٧.                controller_array_operator[i]--;
٦٧٨.              }
٦٧٩.            }
٦٨٠.            pulse_generator();
٦٨١.          }
٦٨٢.        }
٦٨٣.
٦٨٤.        void controller_start()
٦٨٥.        {
٦٨٦.          controller_array[0]= 127;
٦٨٧.          controller_array[1]= 127;
٦٨٨.          controller_array[2]= 0;
٦٨٩.          controller_array[3]= 127;
٦٩٠.
٦٩١.          roll.attach(22);
٦٩٢.          pitch.attach(24);
٦٩٣.          throttle.attach(26);
٦٩٤.          yaw.attach(28);
٦٩٥.
٦٩٦.          controllers();
٦٩٧.        }
```