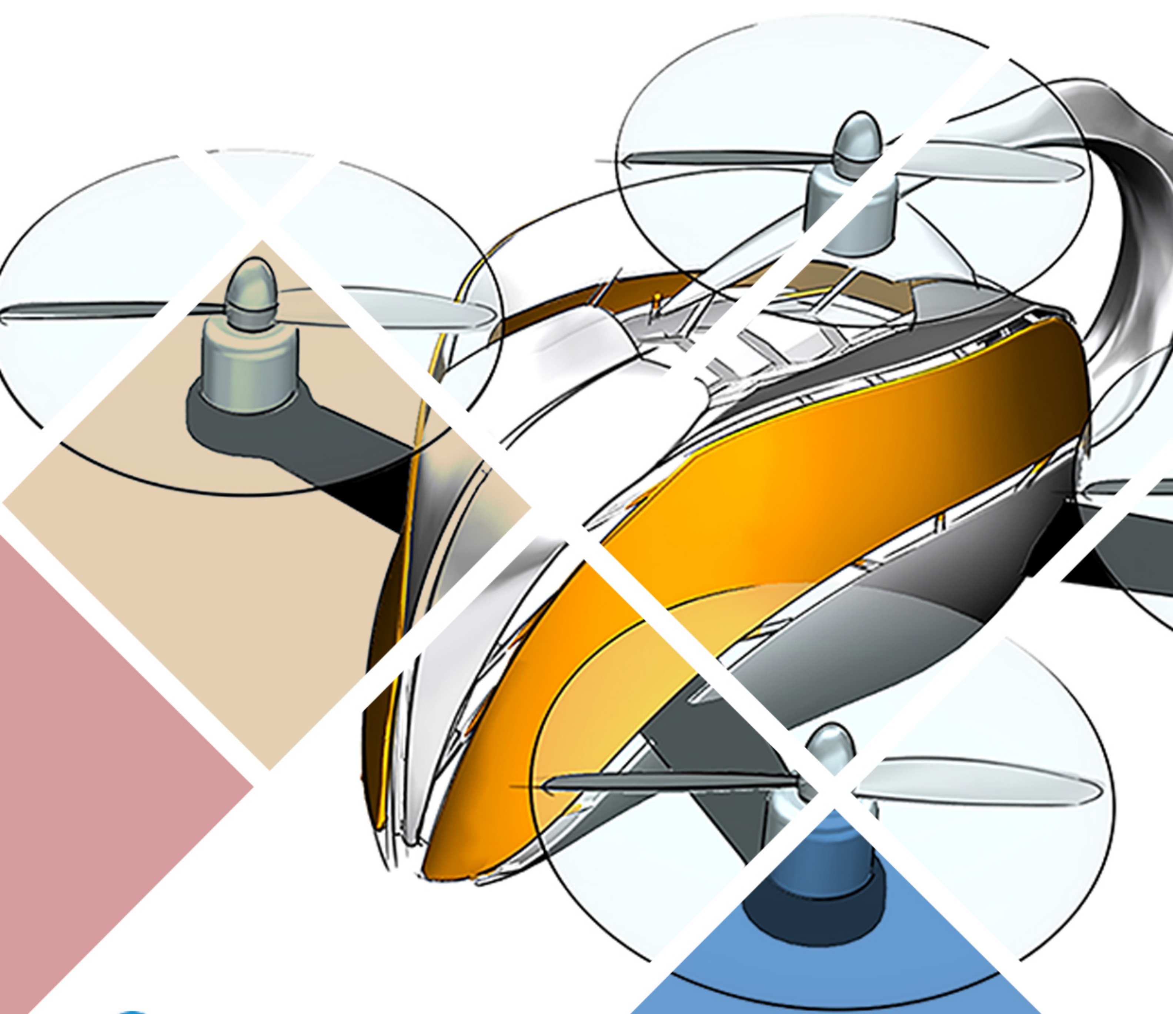


DRONE TEAM

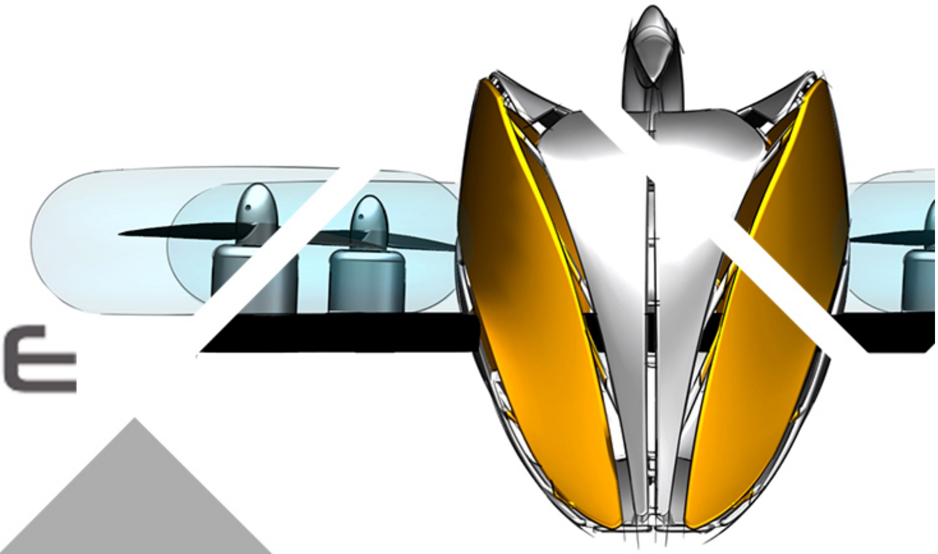
Drone It Yourself!

MAKING AND DESIGNING A TOY DRONE
THROUGH MULTIDISCIPLINARY
COLLABORATIVE WORK
Project no. 2015-1-ES01-KA202-015925

Co-funded by the
Erasmus+ Programme
of the European Union

Drone It Yourself! consists of the following modules:

0. INTRODUCTION TO THE DRONETEAM PROJECT
1. BASIC TOY DRONE FRAME
2. MODULE OF FLIGHT CONTROL
3. MODULE OF COMMUNICATION CONTROL
4. MODULE OF ADVANCED FRAME
5. MODULE OF GPS-COMPASS CONTROL
6. MODULE OF PROBLEM MANAGEMENT
7. MODULE OF FLIGHT STABILIZATION SYSTEM
8. MODULE OF FIRST PERSON VIEW
9. DRONETEAM E-LEARNING PLATFORM
10. OTHER DEVELOPMENTS
11. GLOSSARY



This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

DRONE TEAM

MODULE OF FIRST PERSON VIEW

2015-1-ES01-KA202-015925

Index

1. DronE - DRONE TEAM ANDROID APP

1.1. ANDROID DRONE APP GUIDE

DronE is an application that integrates open source libraries and has been developed so that it can be used and adapted to any drone. It is possible to download it from Google Play. This course explains how to use and configure it. In addition, since it is open source, it is possible to modify it according to the needs.

GET IT ON Google play

DronE - Drone Team App

In DronE app you can find many useful tools like:

1. Map - app uses Google Maps so you can find your position, check landform, and much more. Maps in app are faster and more configurable than in official Google's app. Just check!
2. Add your drone - you can add and show drones and their parts. You can choose between general description or all parts description. After that you will see your and public drones.
3. Countries - check all teams in our project. Read in English or native language.
4. Dark or light theme - Sunny day or dark room. For this app is not a problem. Simply switch allows you to change version.
5. More - check our GitHub or contact, you can also read something more about this project.

1.2. USING ANDROID STUDIO

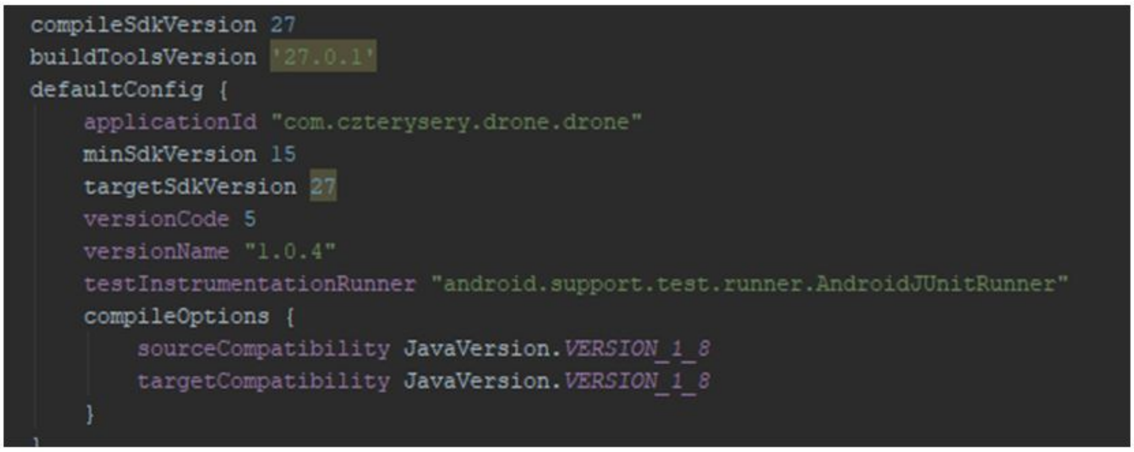
To start working with the app or create an alternative you need to download an appropriate IDE (Android Studio, Eclipse).

android studio

Android Studio is used for developing apps for Android devices, providing tools for building these apps.

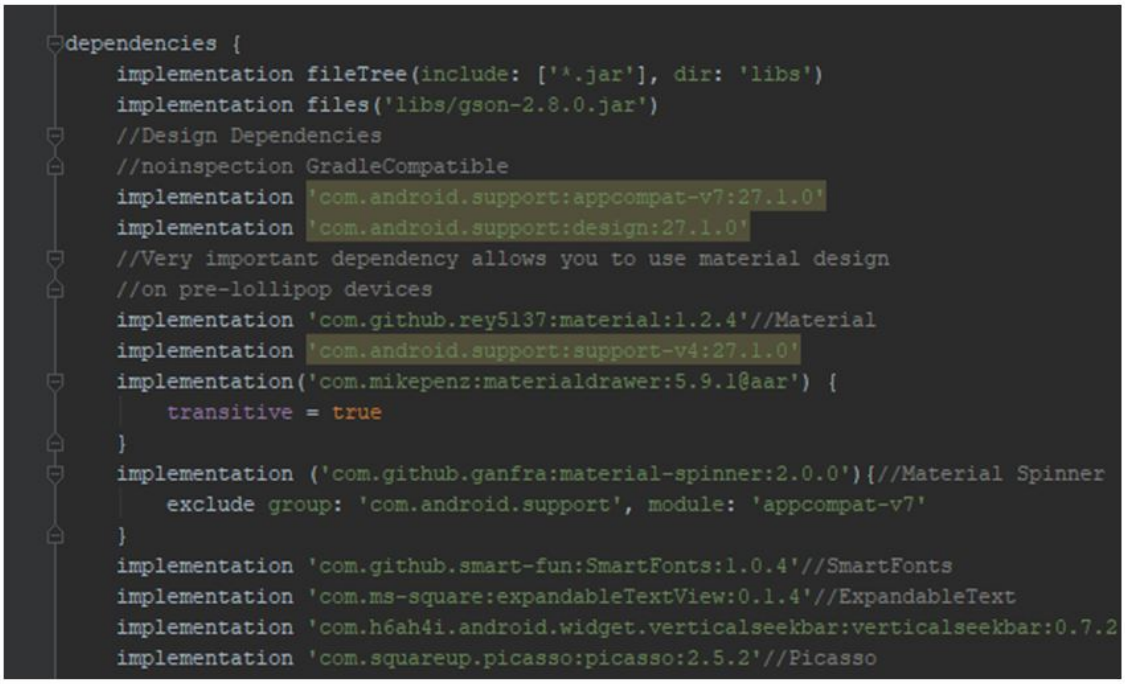
When you download a software and open the app you should go first to the build. gradle file. This file is responsible for two things:

- Defining a target Android system version and SDK (software development kit)
- Importing dependencies and libraries to the app.



```
compileSdkVersion 27
buildToolsVersion '27.0.1'
defaultConfig {
    applicationId "com.czterysery.drone.drone"
    minSdkVersion 15
    targetSdkVersion 27
    versionCode 5
    versionName "1.0.4"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
```

The configuration can be changed anytime but it has an impact on the compilation.



```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation files('libs/gson-2.8.0.jar')
    //Design Dependencies
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:27.1.0'
    implementation 'com.android.support:design:27.1.0'
    //Very important dependency allows you to use material design
    //on pre-lollipop devices
    implementation 'com.github.rey5137:material:1.2.4'//Material
    implementation 'com.android.support:support-v4:27.1.0'
    implementation('com.mikepenz:materialdrawer:5.9.1@aar') {
        transitive = true
    }
    implementation ('com.github.ganfra:material-spinner:2.0.0'){//Material Spinner
        exclude group: 'com.android.support', module: 'appcompat-v7'
    }
    implementation 'com.github.smart-fun:SmartFonts:1.0.4'//SmartFonts
    implementation 'com.ms-square:expandableTextView:0.1.4'//ExpandableText
    implementation 'com.h6ah4i.android.widget.verticalseekbar:verticalseekbar:0.7.2
    implementation 'com.squareup.picasso:picasso:2.5.2'//Picasso
```

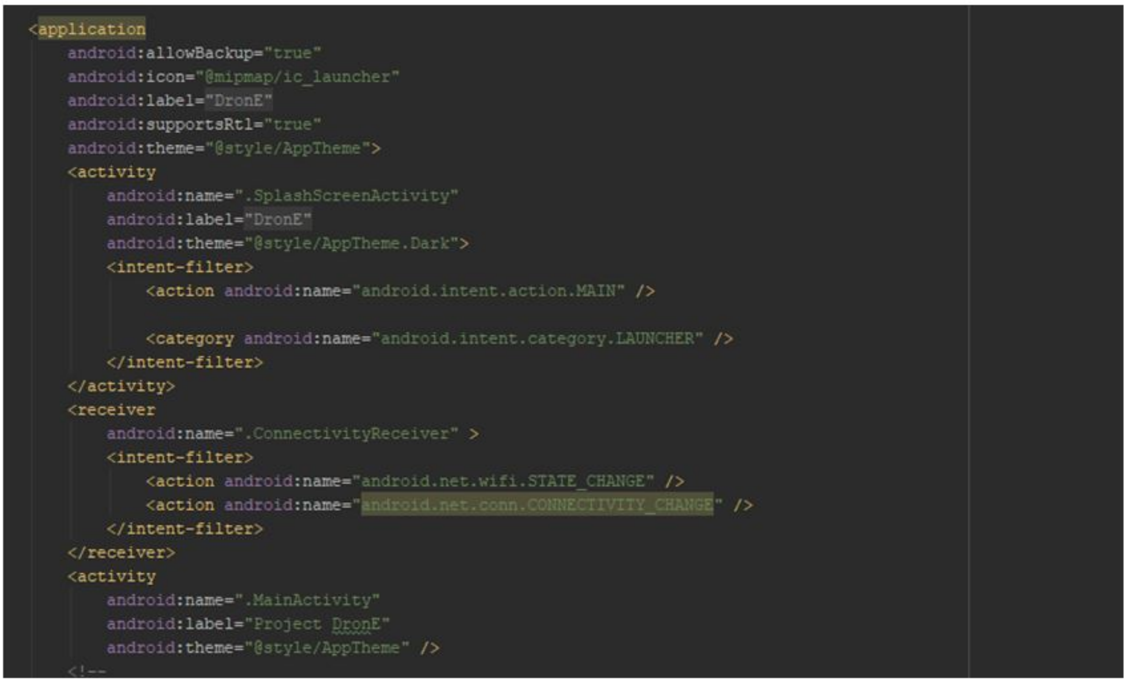
This app uses a big amount of libraries and public dependencies that help contribute project better and faster. DronE uses two bigger technologies:

- **Firebase** - Cloud technology. Allows developers to create databases and follow the statistics without writing any code in the web.

- **Butterknife** - Reduces amount of View-code (like findViewById) using pattern (Dependency Injection)

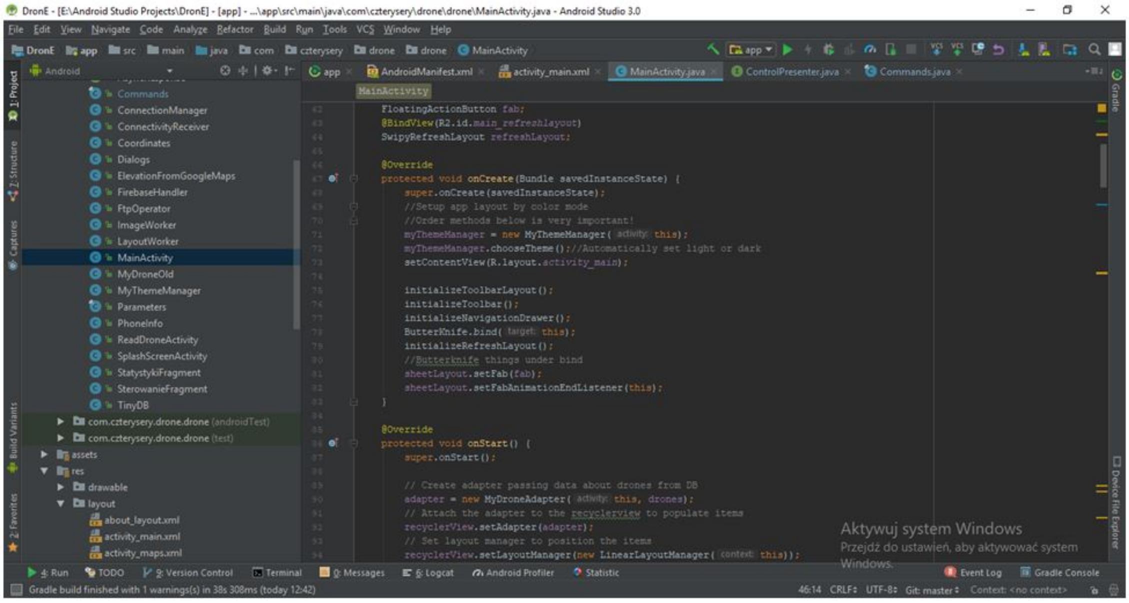
After the all puzzles has been collected for our app we need to organize permissions and rules for it.

In Android app responsibility for this takes file called Manifest.xml. Here we can declare which parts of a system will need permissions like SMS, Contacts, GPS, Camera and we will able to connect classes to the other actions. This is our command center.



```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="DronE"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".SplashScreenActivity"
        android:label="DronE"
        android:theme="@style/AppTheme.Dark">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <receiver
        android:name=".ConnectivityReceiver" >
        <intent-filter>
            <action android:name="android.net.wifi.STATE_CHANGE" />
            <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
        </intent-filter>
    </receiver>
    <activity
        android:name=".MainActivity"
        android:label="Project DronE"
        android:theme="@style/AppTheme" />
    <!--
```

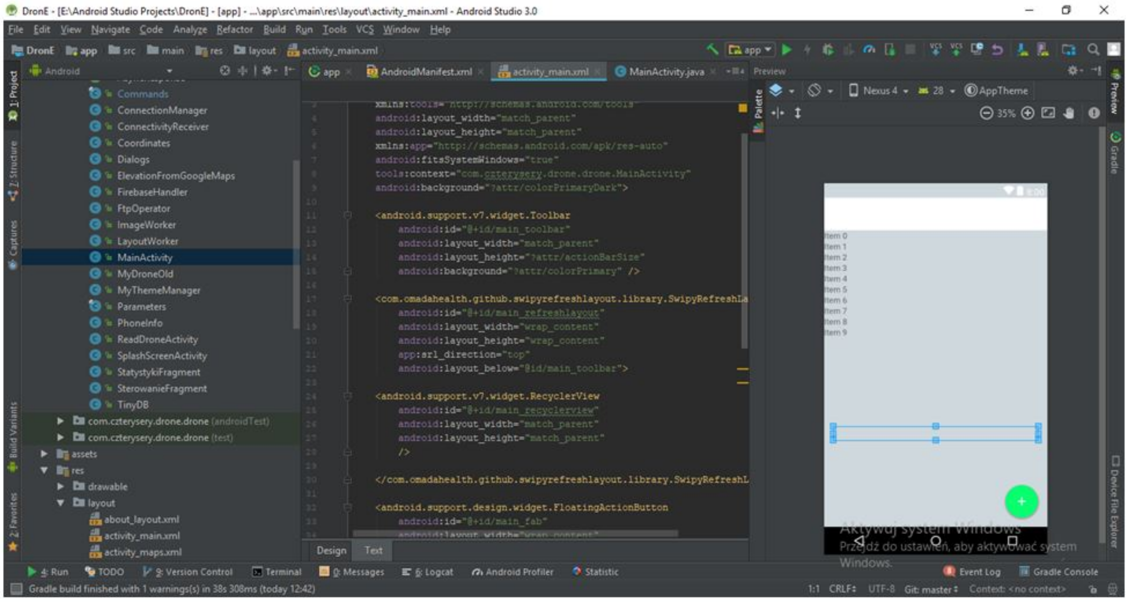
For default main class that starts the app is **MainActivity**. All the code is executed from on Create function. You can see the code on the picture below.



```java
    FloatingActionButton fab;
    @BindView(R2.id.main_refreshlayout)
    SwipyRefreshLayout refreshLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Setup app layout by color mode
        //Order methods below is very important!
        myThemeManager = new MyThemeManager(this);
        myThemeManager.chooseTheme();//Automatically set light or dark
        setContentView(R.layout.activity_main);

        initializeToolbarLayout();
        initializeToolbar();
        initializeNavigationDrawer();
        ButterKnife.bind(this);
        initializeRefreshLayout();
        //Butterknife things under bind
        sheetLayout.setFab(fab);
        sheetLayout.setFabAnimationEndListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();

        // Create adapter passing data about drones from DB
        adapter = new MyDroneAdapter(this, drones);
        // Attach the adapter to the recyclerview to populate items
        recyclerView.setAdapter(adapter);
        // Set layout manager to position the items
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
```

Gradle build finished with 1 warnings(s) in 38s 308ms (today 12:42)

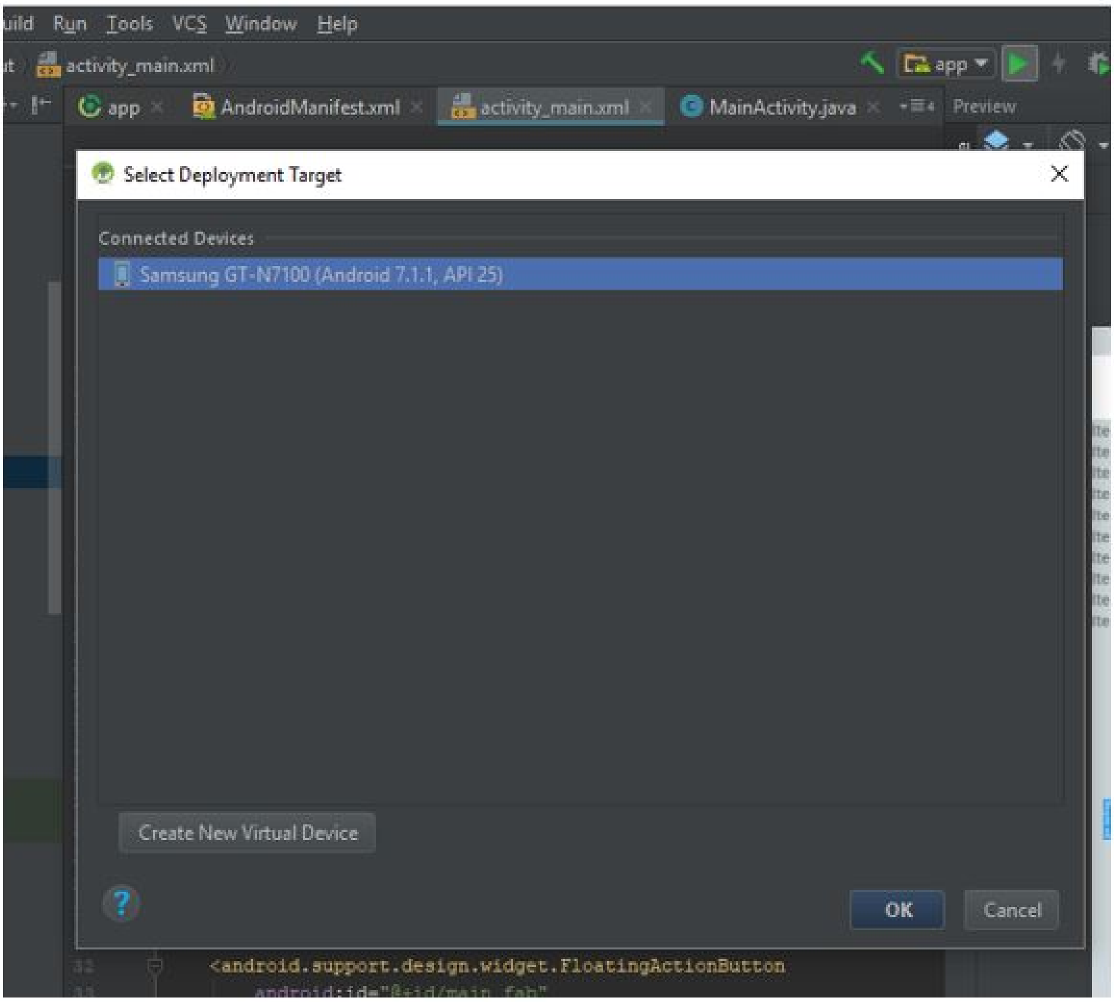
Of course for our app we need some view. The view is invoked by the **setContentView** (R.layout.activity_main); View is nothing more but xml file with the components and their alignment. Android Studio generates a preview so we can see the results before compile action.



```xml
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:fitsSystemWindows="true"
    tools:context="com.czterysery.drone.drone.MainActivity"
    android:background="?attr/colorPrimaryDark">

    <android.support.v7.widget.Toolbar
        android:id="@+id/main_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary" />

    <com.omadahealth.github.swipyrefreshlayout.library.SwipyRefreshLa
        android:id="@+id/main_refreshlayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srl_direction="top"
        android:layout_below="@id/main_toolbar">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/main_recyclerview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        />

    </com.omadahealth.github.swipyrefreshlayout.library.SwipyRefreshL

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/main_fab"
        android:layout_width="wrap_content"
```

Gradle build finished with 1 warnings(s) in 38s 308ms (today 12:42)

If we have all the code and the views, we would check the result. This is possible with app compilation on the real device or an emulator. To compile the app, click the play button (green arrow) and select a device. After successful build app automatically starts on the selected device.



Select Deployment Target

Connected Devices

Samsung GT-N7100 (Android 7.1.1, API 25)

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/main_fab"
```

After the app passed all the tests and we want to publish it on the Google Play Store we have to generate private key (keystore). How to do that you will find at **Android Keystore System Guide (click here to Access: https://developer.android.com/training/articles/keystore)**.

1.3. CHECKING CODE & CONTRIBUTIONS

You can find DronE code at GitHub, a repository used to host open source software normally.

Stakeholders can check the code and contributions always are welcome. Please click here to access to DronE code at GitHub: https://github.com/tmaxxdd/DronE

This is official repository for DronE android application made for using with drones. App uses Butterknife, Unit tests, dependencies and Google services.

2. NEW COMPONENTS

2.1. NEW COMPONENTS TO ASSEMBLY LAST PHASE FOR ADVANCED DRONE:

Components for FPV (First Person View):

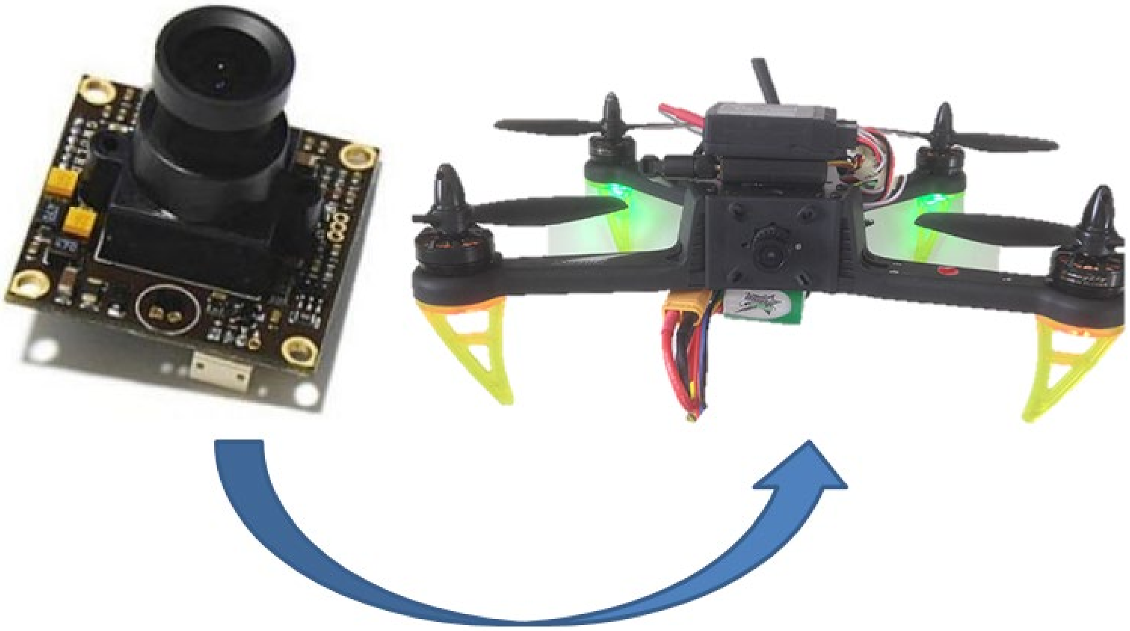
- Mini Camera
- FPV Monitor 7”
- Transmitter and video receiver
- OSD

Mini OSD V2.1 protocol MAVLink Compatible with APM

https://rc-innovations.es/Mini-OSD-protocolo-MAVLink

OSD (On Screen Display) =>Telemetry

- Mini Camera 700TVL Sony CCD

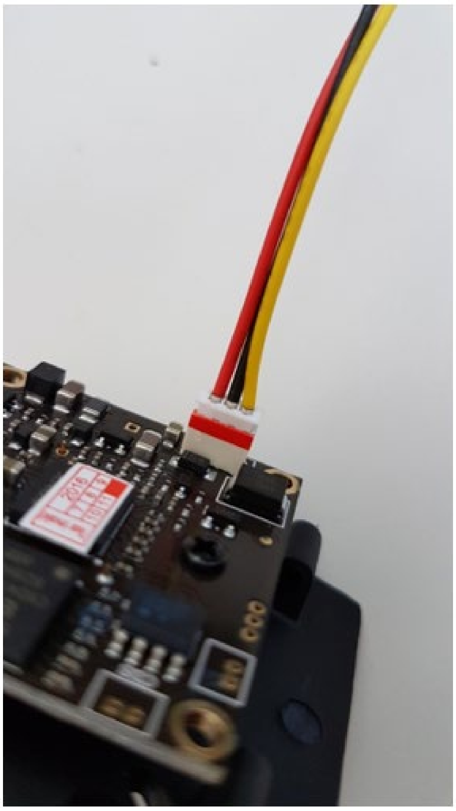


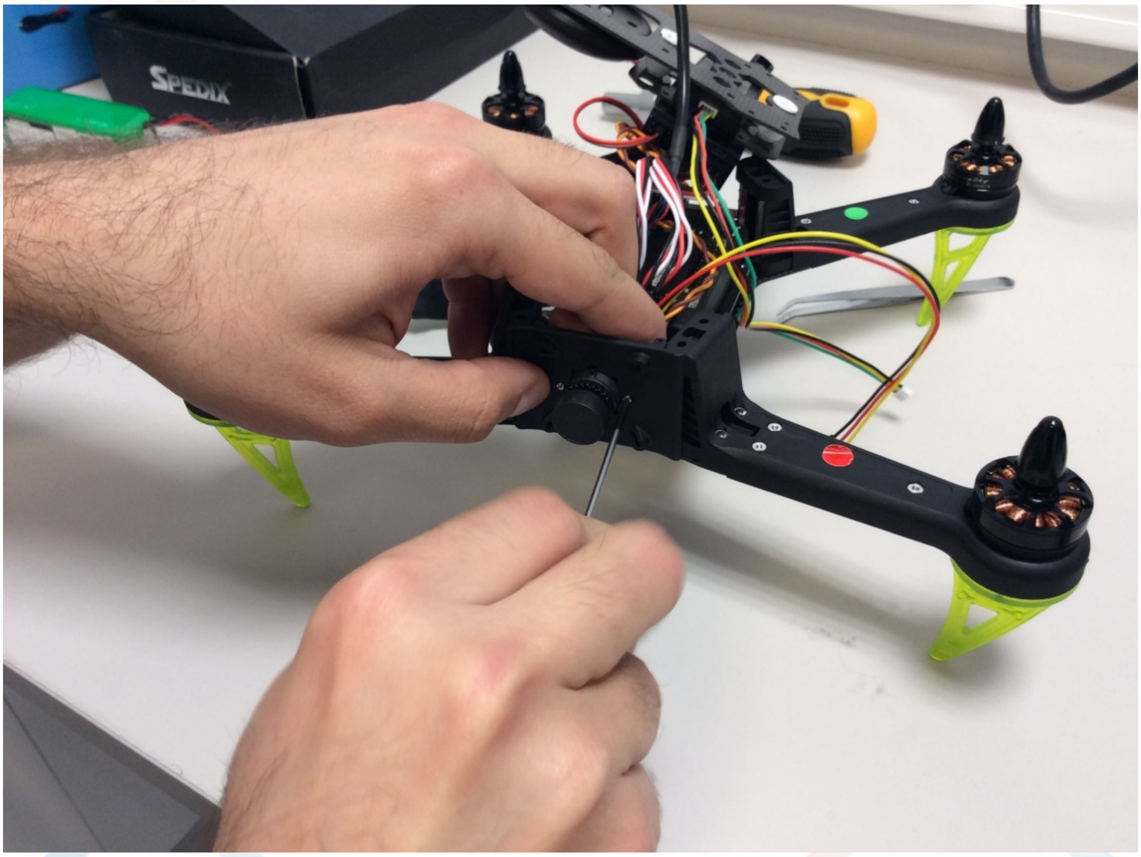
- Monitor FPV Boscam 7" HD diversity

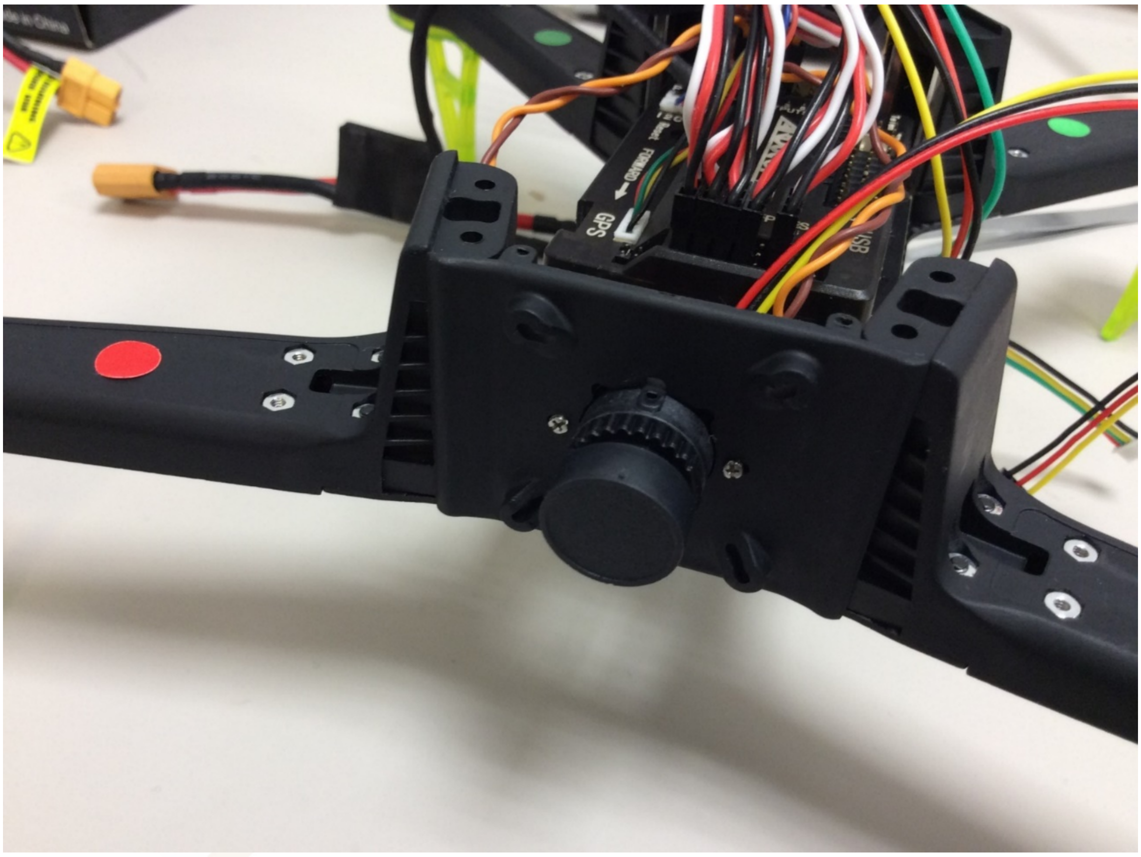
https://rc-innovations.es/Pantalla-7-pulgadas-FPV--Receptor-5.8Ghz-dual-diversity-bateria-interna-grabador-tarjeta-sd-cargador-

- Kit Tx/Rx video 5.8Ghz 200mW

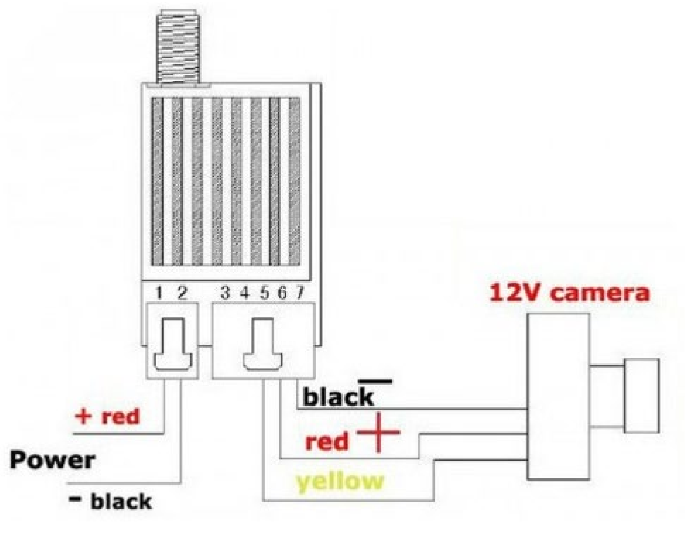
https://rc-innovations.es/Kit-transmisor-receptor-video-5.8Ghz-200mW-rc-305-

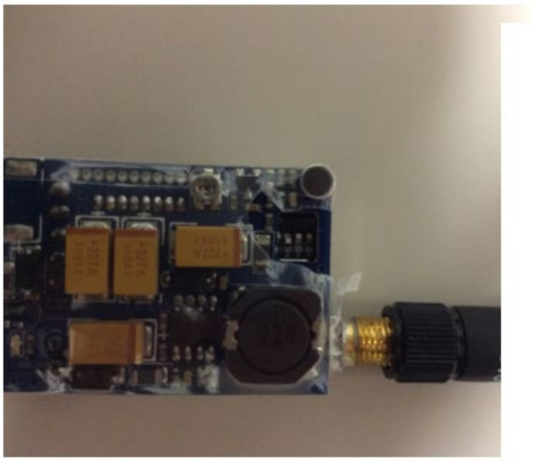


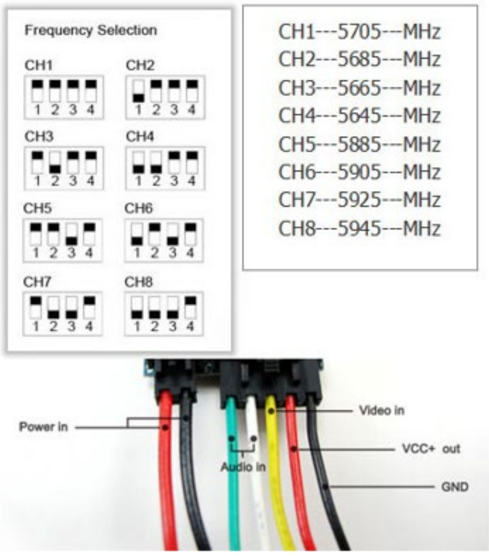

SPEDIX



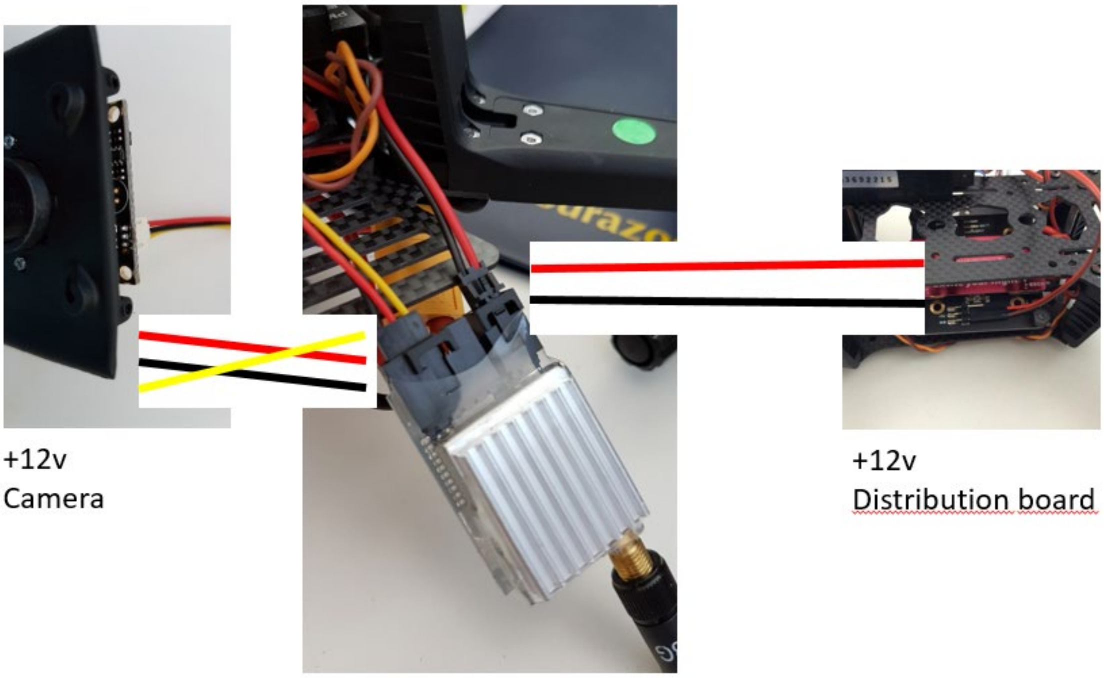
2.2. VIDEO TRANSMMITTER







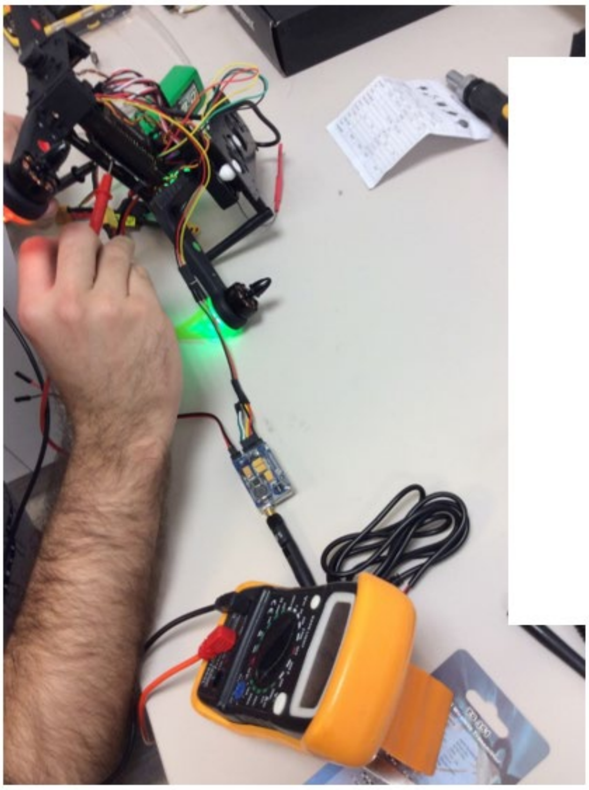
CH1---5705---MHz
CH2---5685---MHz
CH3---5665---MHz
CH4---5645---MHz
CH5---5885---MHz
CH6---5905---MHz
CH7---5925---MHz
CH8---5945---MHz

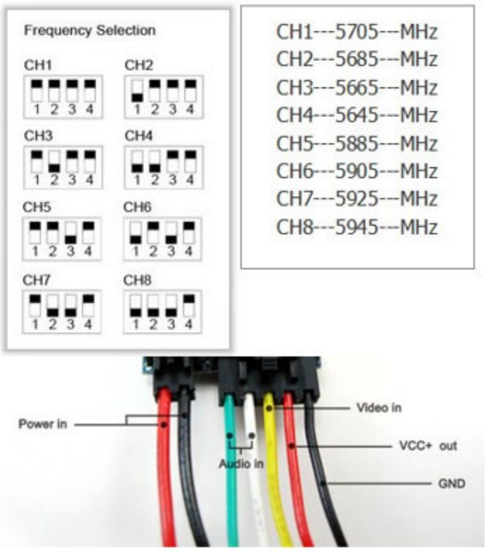


+12v
Camera

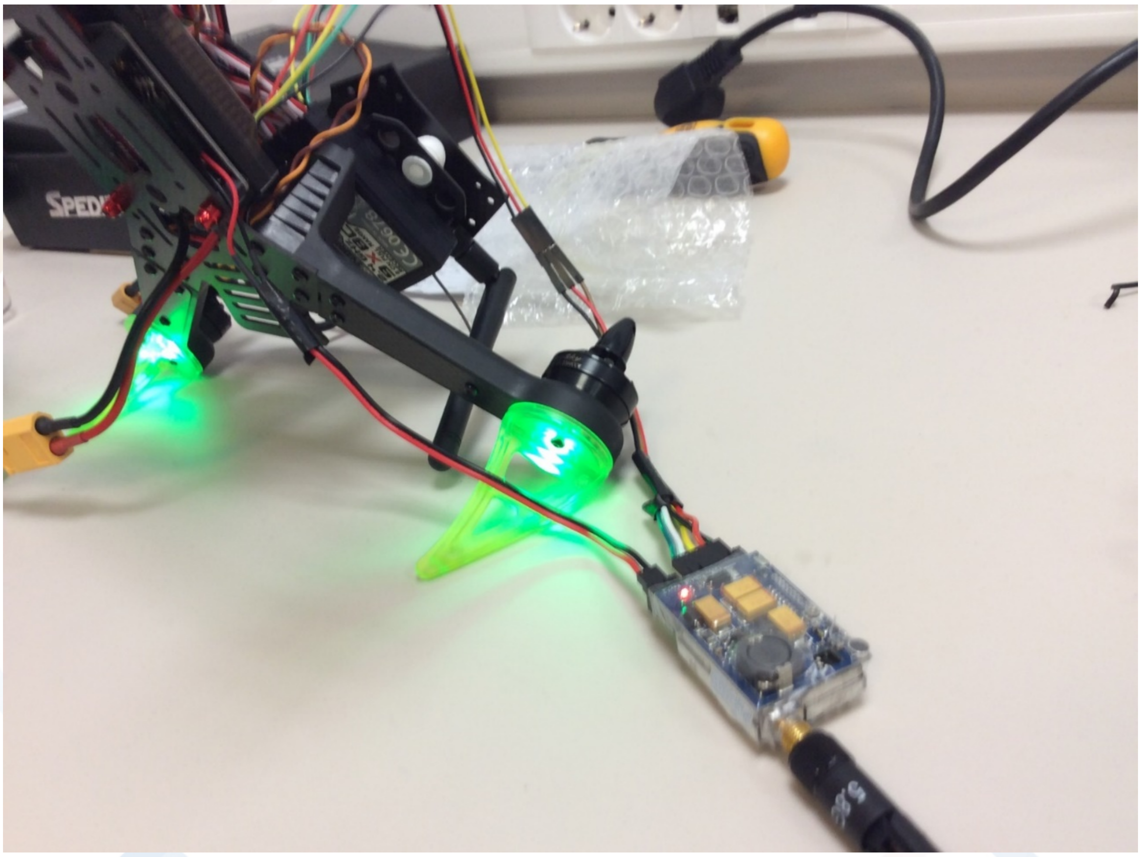
+12v
Distribution board

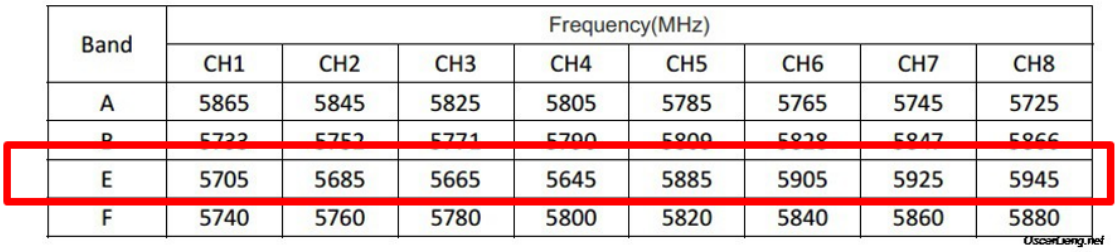
Video transmmitter





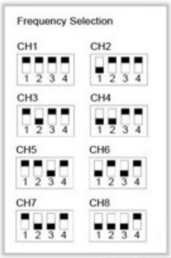
CH1---5705---MHz
CH2---5685---MHz
CH3---5665---MHz
CH4---5645---MHz
CH5---5885---MHz
CH6---5905---MHz
CH7---5925---MHz
CH8---5945---MHz

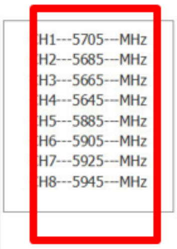




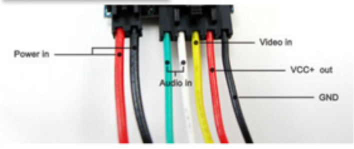
Band	Frequency(MHz)							
	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
A	5865	5845	5825	5805	5785	5765	5745	5725
B	5733	5752	5771	5790	5809	5828	5847	5866
E	5705	5685	5665	5645	5885	5905	5925	5945
F	5740	5760	5780	5800	5820	5840	5860	5880

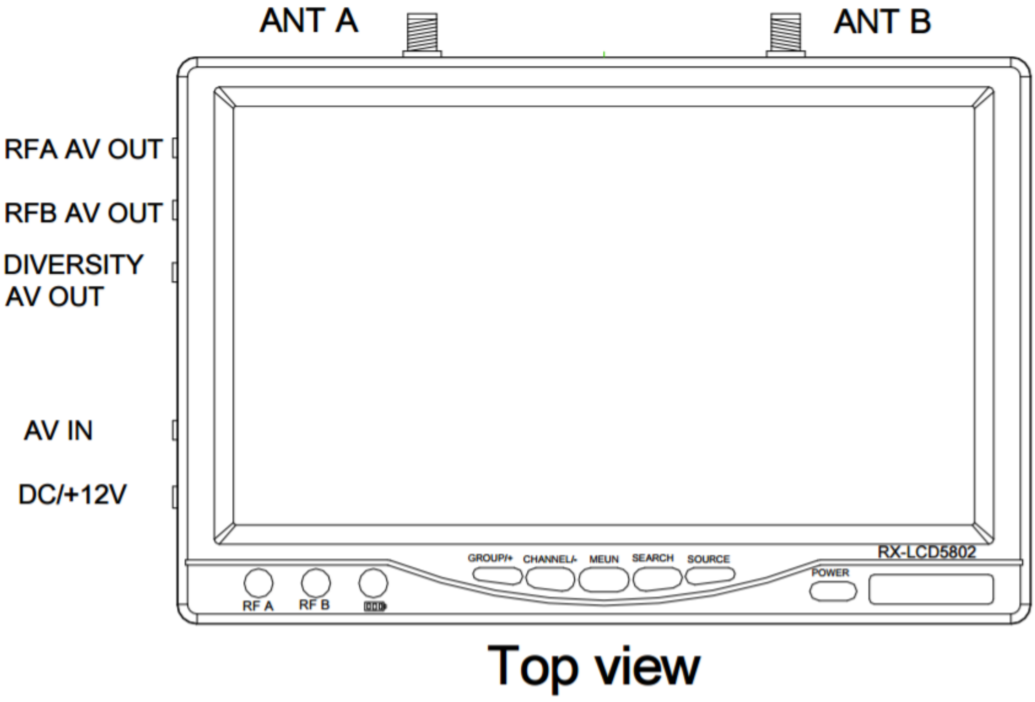
OscarLiang.net





H1---5705---MHz
H2---5685---MHz
H3---5665---MHz
H4---5645---MHz
H5---5885---MHz
H6---5905---MHz
H7---5925---MHz
H8---5945---MHz





1) GROPU/+：

Default mode，click the GROUP option，A, b, c, d cycle selection；MEUN pressed，this button is used as number +；

2）CHANNEL/-：

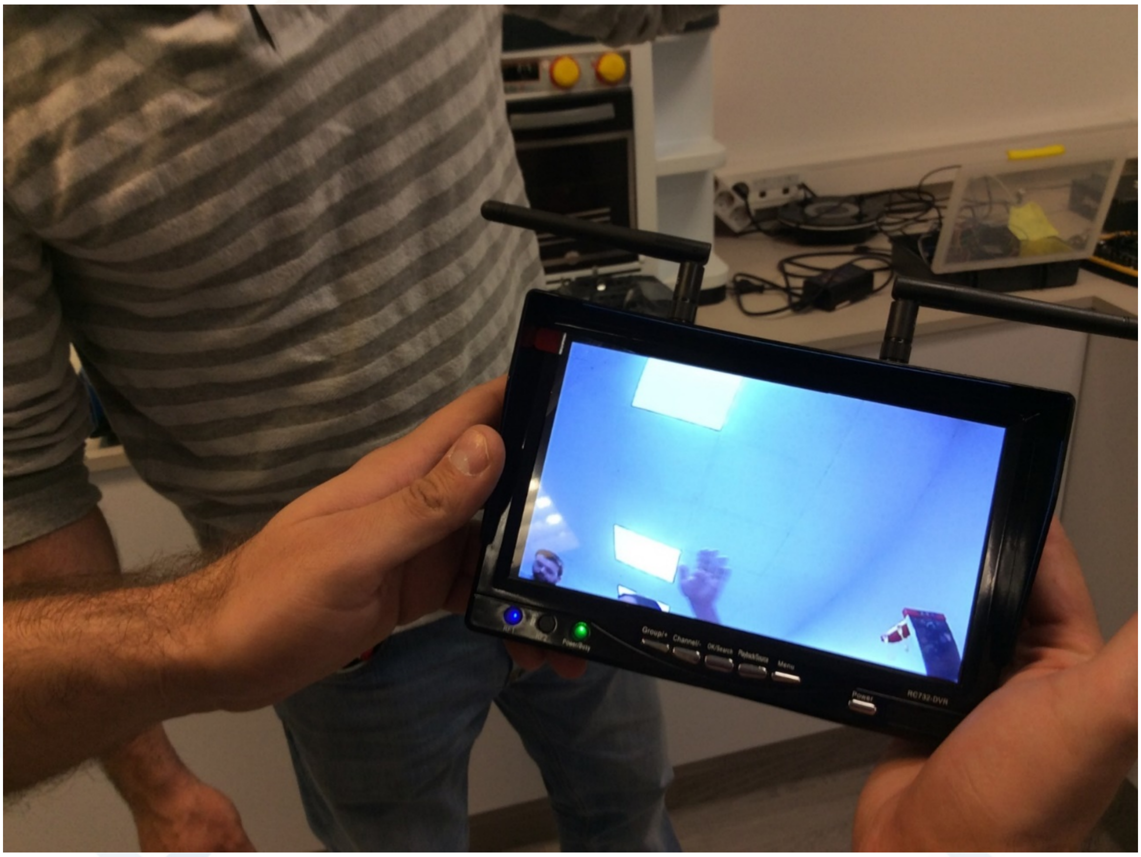
Default mode，click the option for the CH 1-8 cycles；MEUN pressed，this button is used as the value -；

3）MEUN：

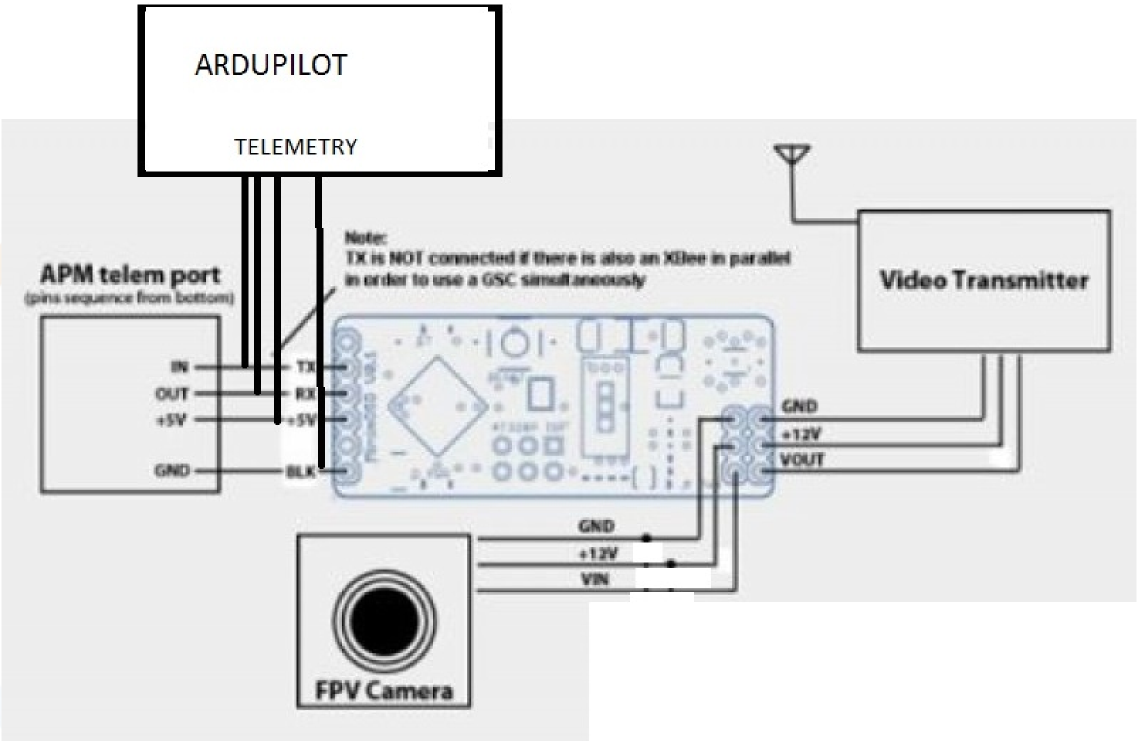
LCD display parameter adjustment，where the parameters using +，- to adjust，click MEUN, choose a different adjustment items；

4) SEARCH：

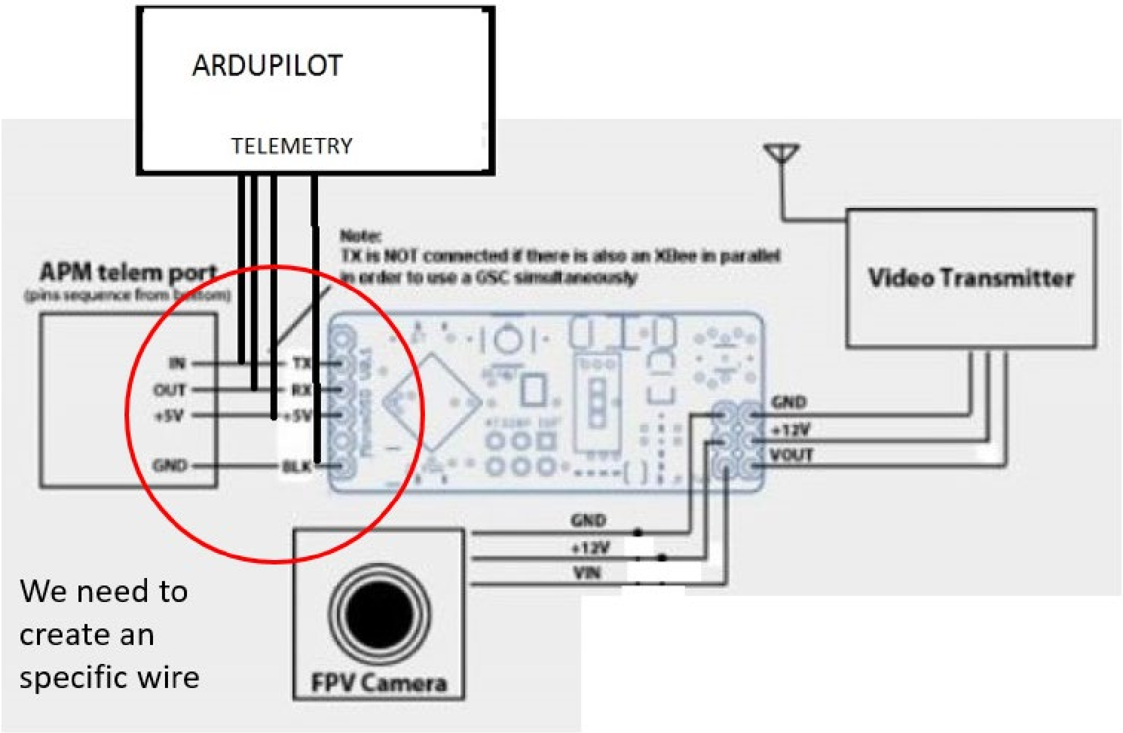
Press this button to enter the signal search，it will automatically search for the signals to a certain signal strength and stops on that channel，the signal is not found，will be on the channel before starting the search stopped；searching process, all the buttons will not operate；



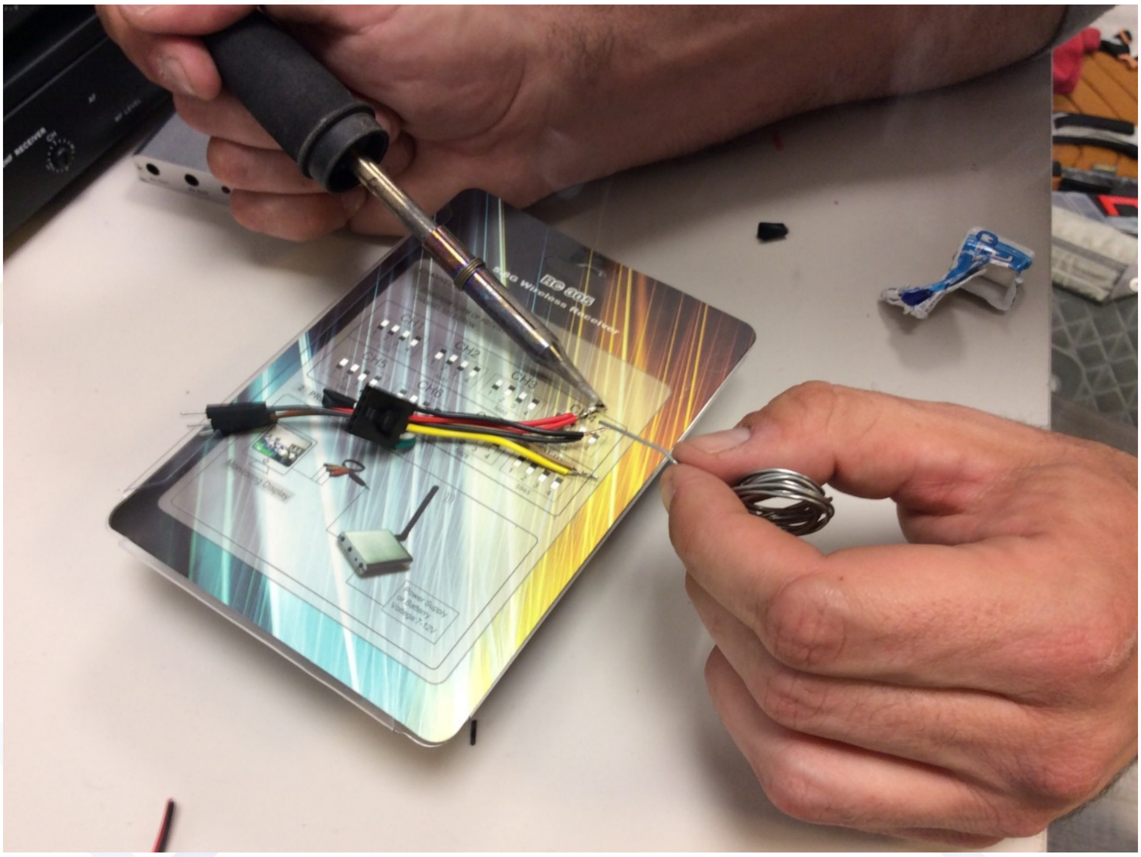
2.3. ADDING TELEMETRY

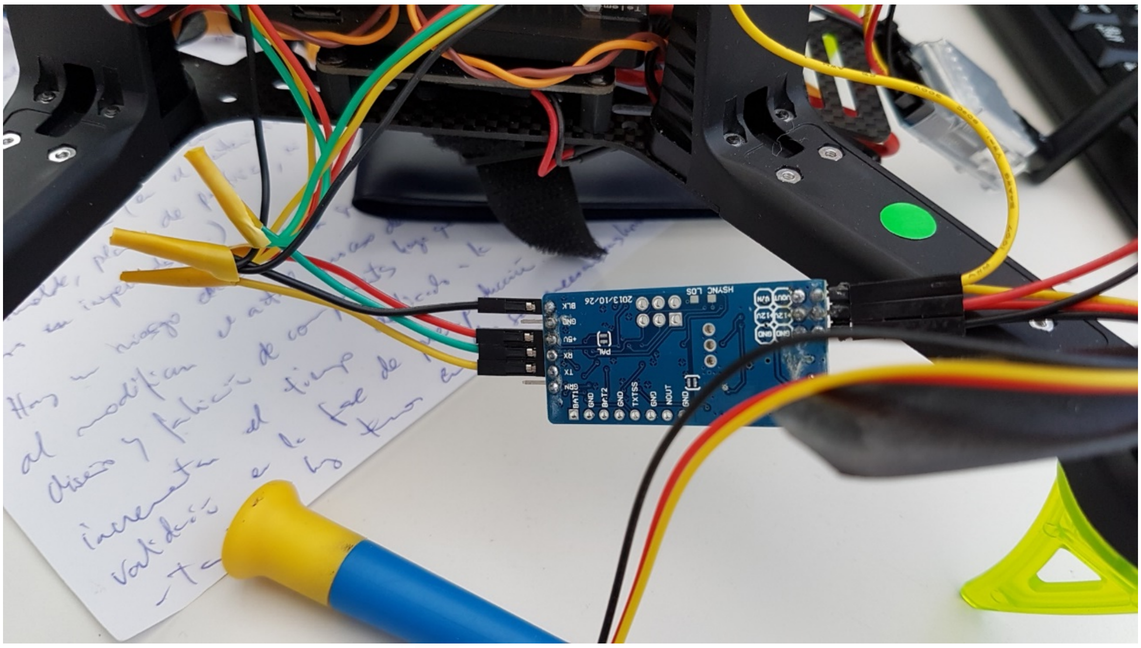


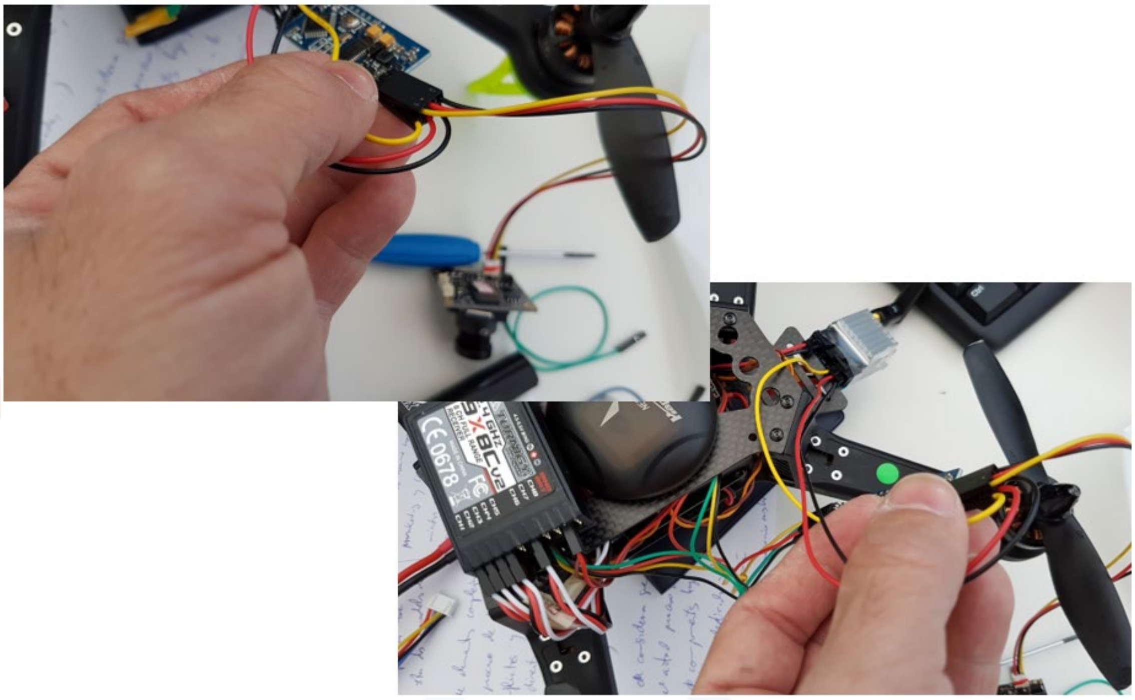
Note:
TX is NOT connected if there is also an XBee in parallel in order to use a GSC simultaneously

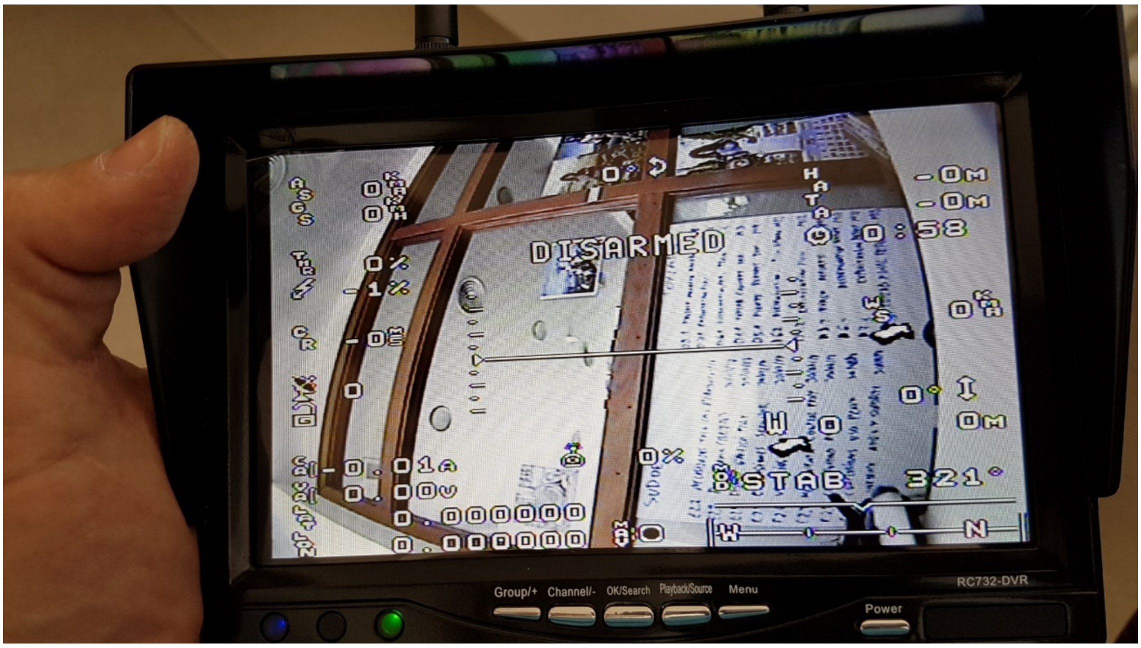

ARDUPILOT
TELEMETRY
APM telem port
(pins sequence from bottom)
IN
OUT
+5V
GND
TX
RX
+5V
BLK
Note:
TX is NOT connected if there is also an XBee in parallel in order to use a GSC simultaneously
Video Transmitter
GND
+12V
VOUT
GND
+12V
VIN
FPV Camera

We need to create an specific wire



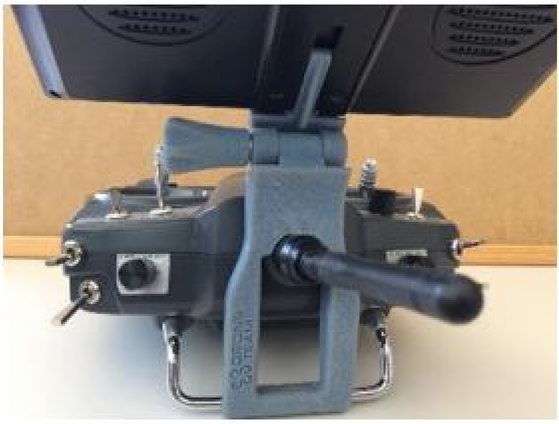

2013/10/26
HSYNC
LOS
PAL
BLK
GND
5V+
RX
TX
GRN
BATT1
GND
BAT2
GND
TXTSS
GND
NOUT
GND
VIN
+12V
VOUT
GND

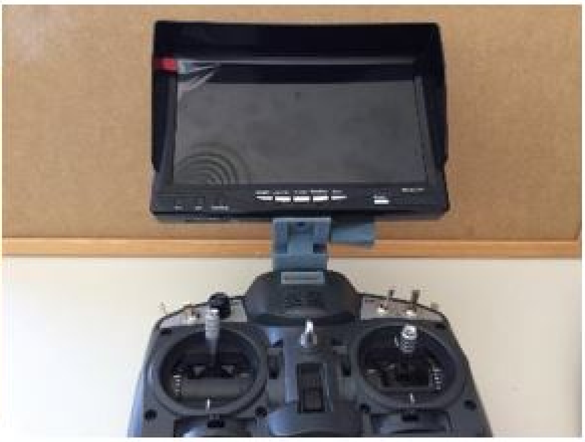


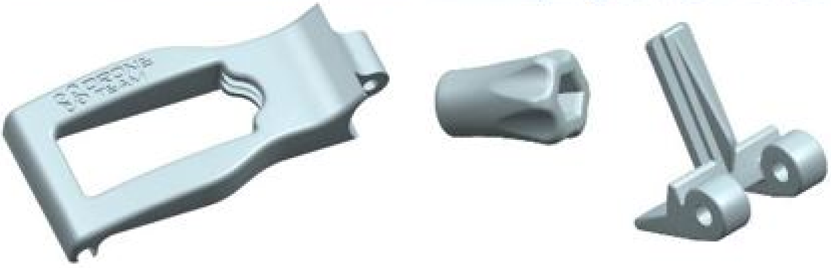


3. CUSTOMIZATION PHASE

Assembly the transmitter and the screen of First Person View. 3D files created and ready for 3D printing. 3 parts, you only need a 5X50mm screw:







Download link: http://www.droneteamproject.eu/results/download-area2/223-support-screen-fpw

Other options for customising: it is advisable to adapt a suitcase to transport our drone. Keep in mind that batteries can cause problems in the airports if they are deposited inside the checked bags.

Sometimes we do not have the possibility or the time to use a 3D printer to replace a broken part. Creativity and being a maker can solve the problem, although it is not aesthetic, it is functional and does not have an additional weight: we use a rubber band used in a chair as a leg for our drone!