

Seminár Robotika.SK

Mobilný telefón ako palubný počítač na robotovi

Richard Balogh

*Katedra automobilovej
mechatroniky FEI STU*

balogh@elf.stuba.sk

<http://aladin.elf.stuba.sk/~balogh>

Andrej Lúčný

*Katedra aplikovanej
informatiky FMFI UK*

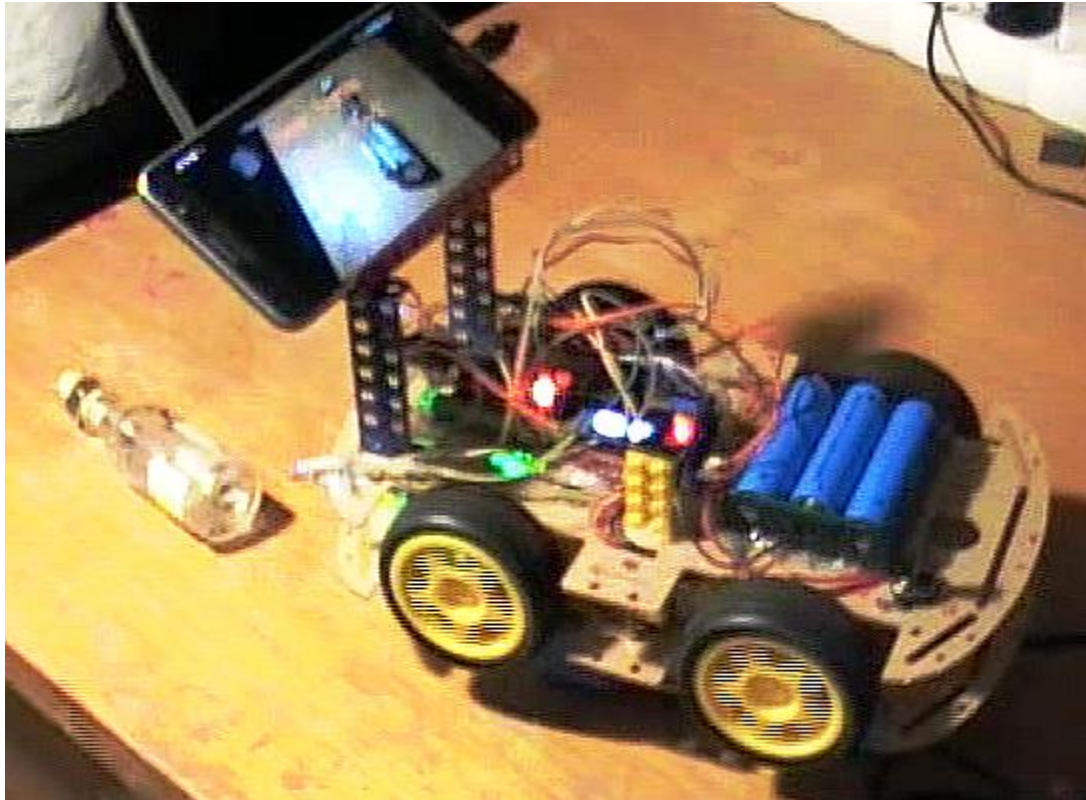
lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny

www.robotika.sk/seminar/2021/robotikask-openbot.pdf

Mobilný telefón ako súčiastka

Výhody: dostupný, malý, výkonný počítač s batériou a kamerou



Zapojenie

OpenBot
Firmware

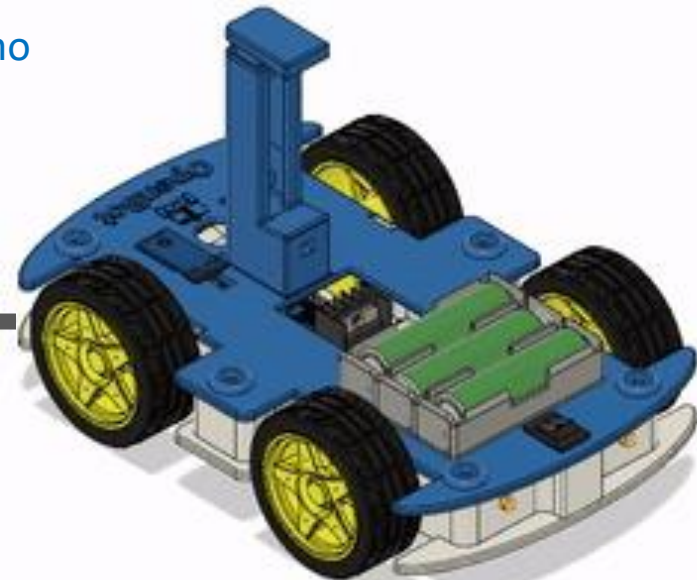
`openbot_v1_nano.ino`



smartphone
Android



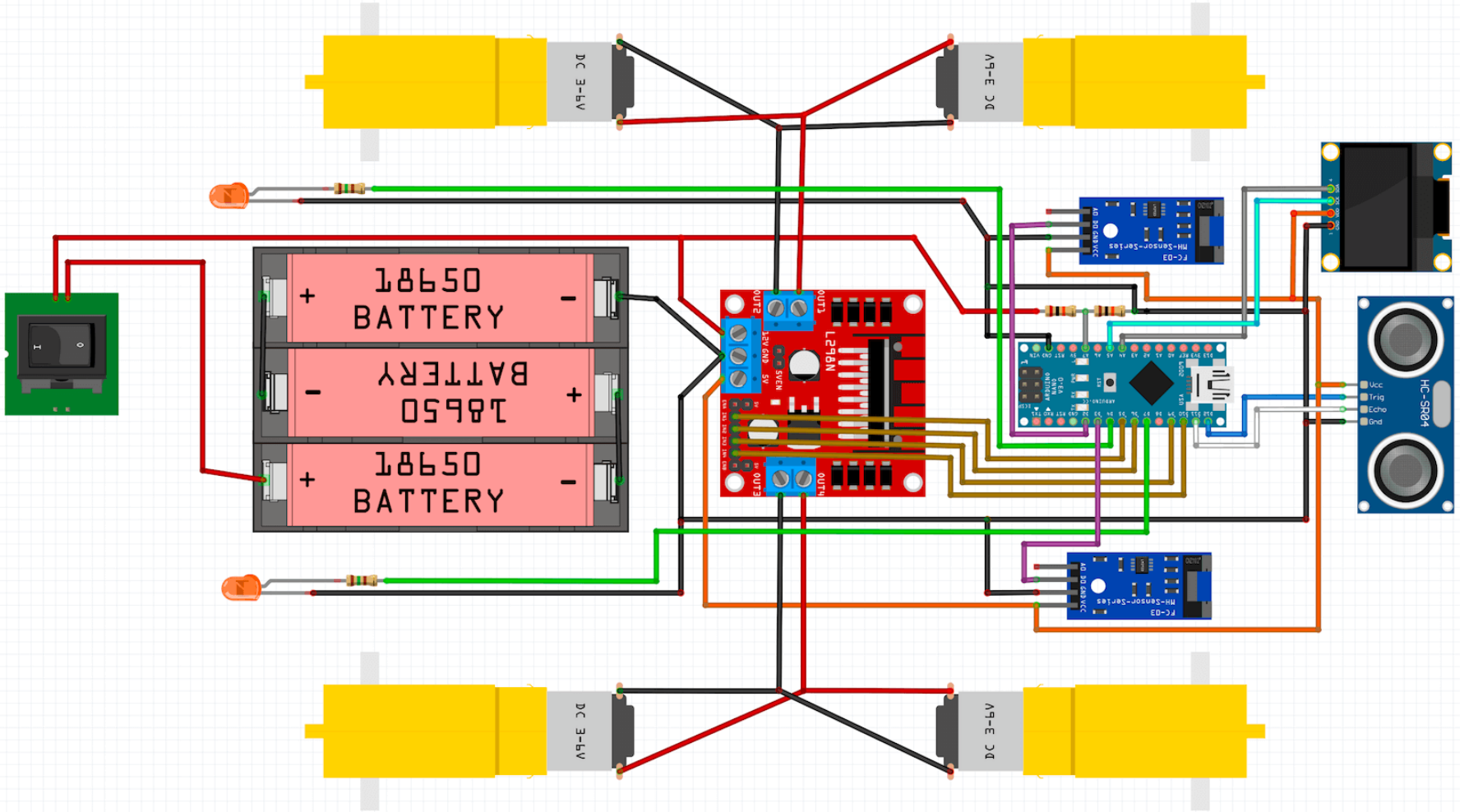
Arduino
Nano



OpenBot

www.openbot.org

OpenBot



OpenBot - napájanie

Napájanie je trocha zvláštne vyriešené:

- Arduino je napájané cez USB kábel z mobilu
 - je to tak kvôli tomu, že ak je Arduino napájané z externého zdroja, FTDI chip nie je napájaný a komunikácia cez USB by nefungovala.
 - pozor! zd'aleka nie každý kábel k mobilu podporuje aj komunikáciu aj napájanie z mobilu – to musí byť tzv. OTG kábel
- všetko ostatné je na baterky 9V, ktoré sa zapnú vypínačom

OpenBot - protokol

Rýchlosť sériovej linky je 115 200 Baud.

OpenBot posiela 1x za sekundu štyri čísla oddelené čiarkou:

NapatieBaterie, rychlostL, rychlostP, ultrazvukovySenzor<CR><LF>

Ultrazvukovy senzor meria v cm, ako ďaleko je k najbližšej prekážke. Ak má nejakú poruchu, vracia 65535

OpenBot - protokol

Príkazy, ktoré sa dajú po sériovej linke posielat', sú:

i1<CR><LF>

ľavý blinker

i-1<CR><LF>

pravý blinker

i0<CR><LF>

vypni svetielka

c255, 255<CR><LF>

max. rýchlosť vpred

c-225, -255<CR><LF>

max. rýchlosť vzad

c0, 0<CR><LF>

stop

c127, -127<CR><LF>

otočka

Ako naprogramovať aplikáciu spracúvajúcu obraz pod Android



(vychádza z DNN-OpenCV-Classification-Android na [learnopencv.com](https://github.com/smallick/learnopencv.git) <https://github.com/smallick/learnopencv.git>)

Vývojové prostriedky

developer.android.com/studio

<https://opencv.org/releases>



android



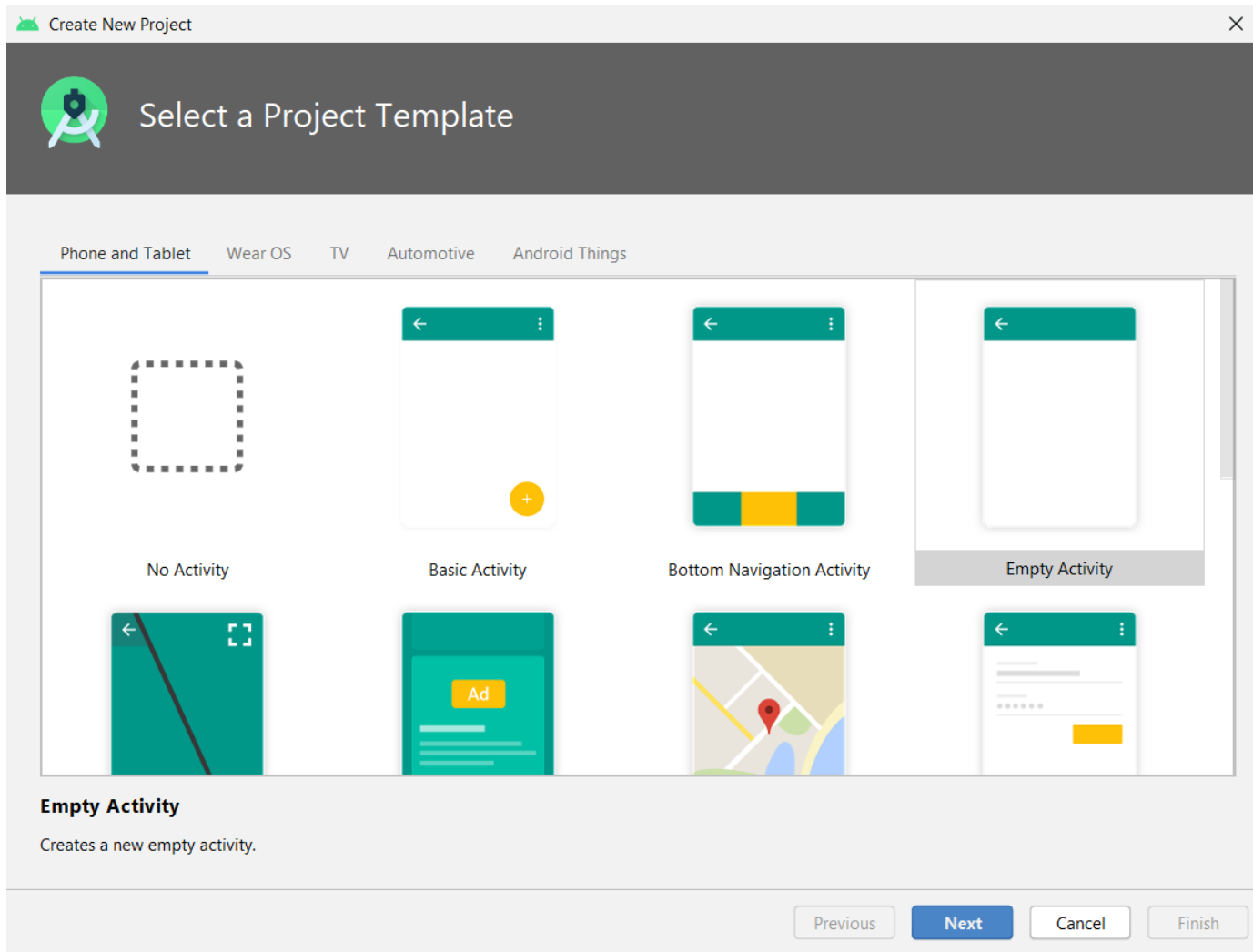
4.0.1-android-studio-ide-
193.6626763-windows.exe



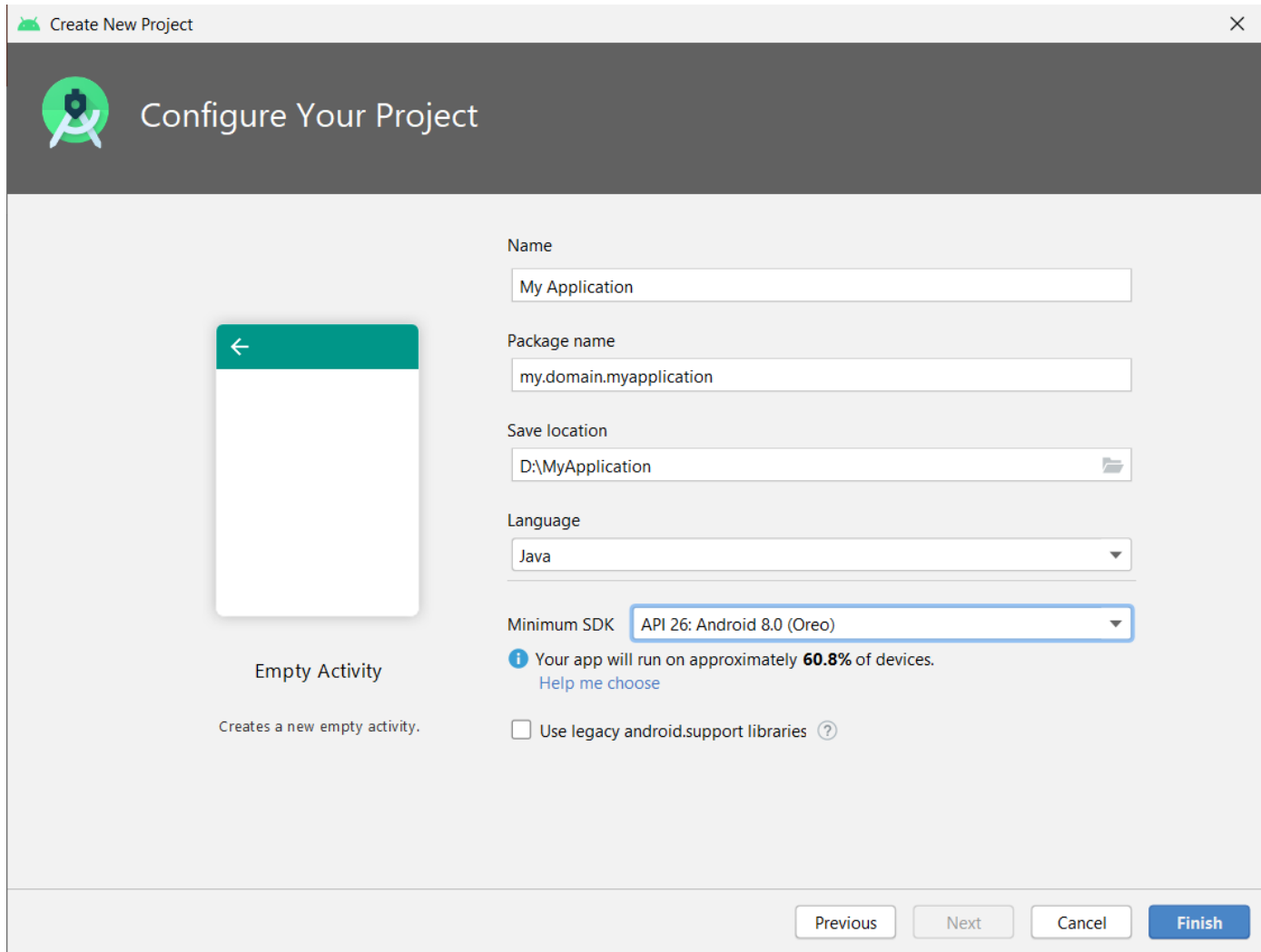
OpenCV

pre Android SDK

opencv-4.5.0-android-sdk.zip
[c:/opencv450](#) /OpenCV-android-sdk



V Android studio začneme projekt z Empty Activity



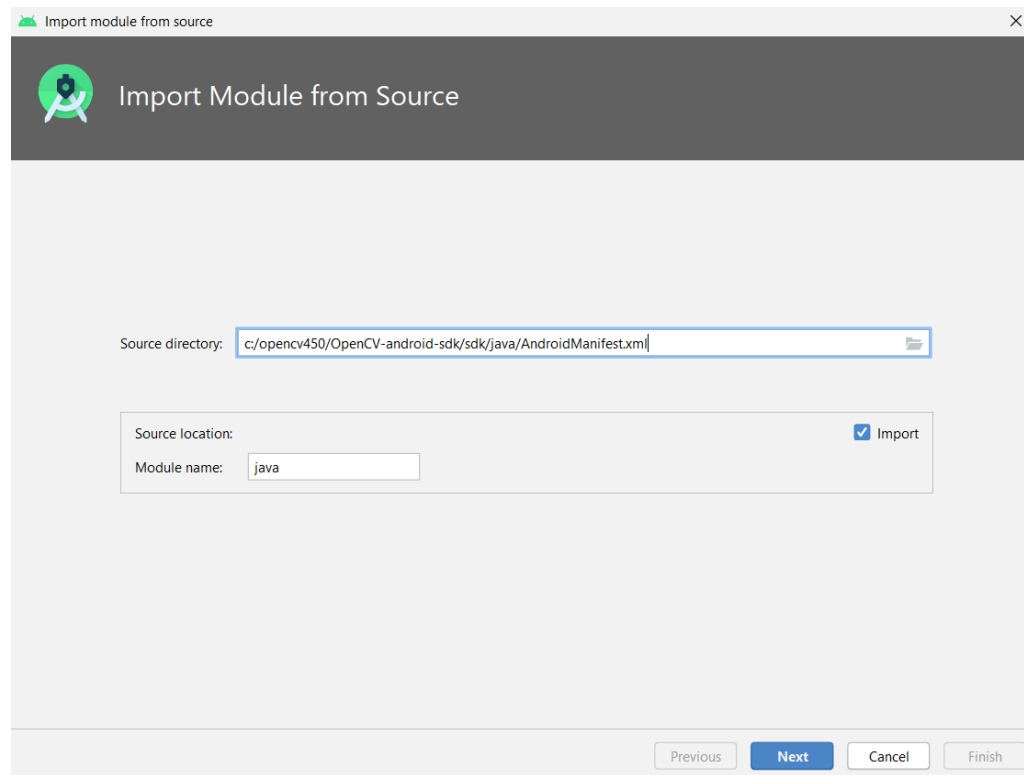
zadáme cesty projektu a vytvoríme ho

- File / New / import-module

`c:/opencv450/OpenCV-android-sdk/sdk/java/AndroidManifest.xml`

module name: java → OpenCVLib450

v adresári projektu vznikne adresár OpenCVLib450



Do projektu vložíme OpenCV

```
apply plugin: 'com.android.application' ← library

android {
    compileSdkVersion 26 29 ← z app/build.gradle
    buildToolsVersion "30.0.2" ← zkopírujeme vezrie
                                   SDK a buildTools

    defaultConfig {
        applicationId "org.opencv"
        minSdkVersion 16
        targetSdkVersion 29
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile(
                'proguard-android.txt'), 'proguard-rules.txt'
        }
    }
}
```

upravíme `OpenCVLib450/build.gradle` + sync

```
...
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-andro...
    }
}
}

dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.localbroadcastmanager:localbroadcast...
    implementation files('libs\\usbserial.jar')
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-co...
    implementation project(':OpenCVLib450') ← pridáme
}

```

upravíme `app/build.gradle` a skompilujeme Ctrl-F9

[3] Total Commander (x64) 8.01 - Andrej Lucny

Files Mark Commands Net Show Configuration Start Help

FTP Transfer mode Binary (archives, doc) Disconnect Rename to: \\var\www\vhosts\www.agentspace.org\html\opencv\opencv430-ouda102.zip
Get directory: \\var\www\vhosts\www.agentspace.org\html\opencv

c [windows] 37,211,020 k of 484,336,636 k free \ .. c [windows] 37,211,020 k of 484,336,636 k free \ ..

c:\opencv450\OpenCV-android-sdk\sdk\native\libs*.*

Name	Ext	Size	Date
[..]	<DIR>		11/21/2020
[arm64-v8a]	<DIR>		11/21/2020
[armeabi-v7a]	<DIR>		11/21/2020
[x86]	<DIR>		11/21/2020
[x86_64]	<DIR>		11/21/2020

copy

c:\andy\projekty\ThirdEye\app\src\main\jniLibs*.*

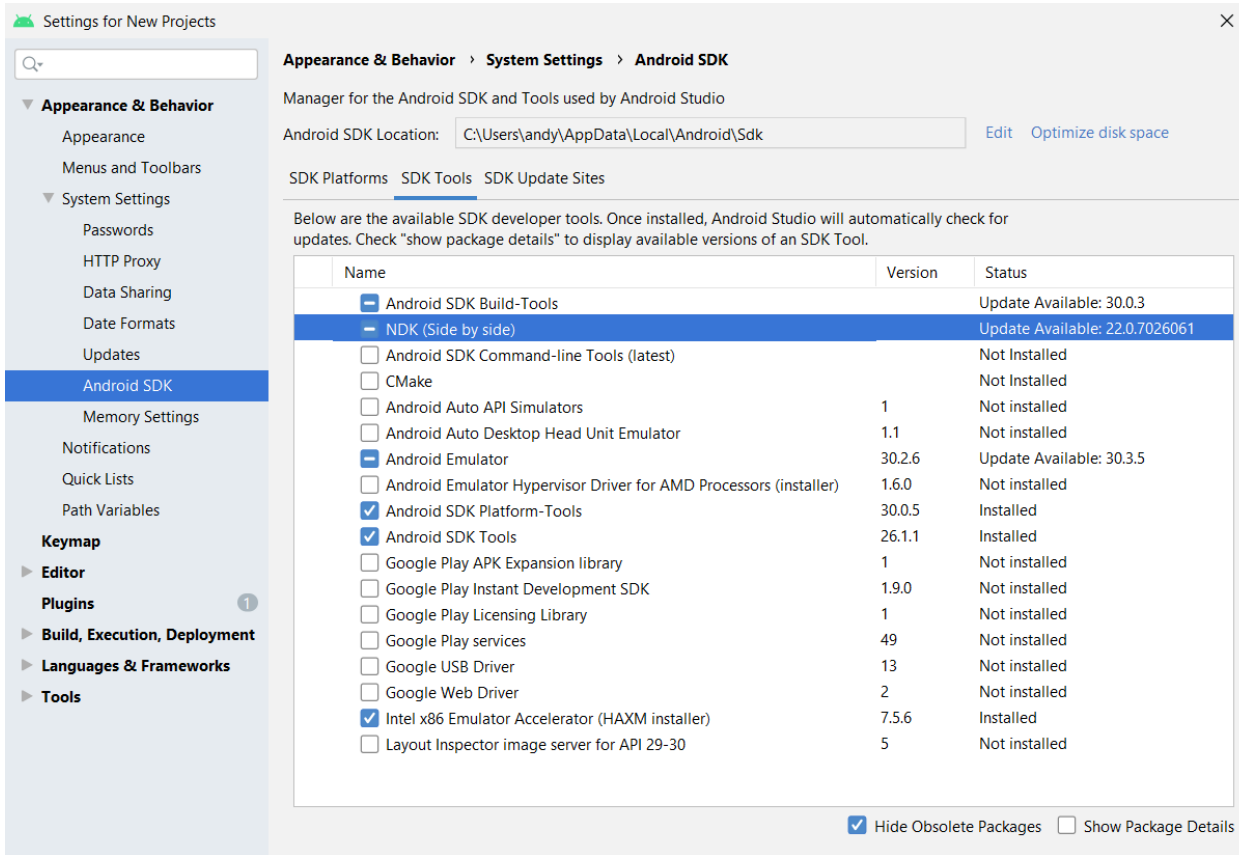
Name	Ext	Size	Date
[..]	<DIR>		11/22/2020
[arm64-v8a]	<DIR>		11/22/2020
[armeabi-v7a]	<DIR>		11/22/2020

modernejší hw
zastarenejší hw

Name	Ext	Size	Date
[..]	<DIR>		11/22/2020
libopencv_java4	so	19,205,584	10/12/2020

Do projektu vytvoríme adresár `app/src/main/jniLibs` a nakopírujeme do neho natívnu knižnicu OpenCV `libopencv_java4.so`

Stačí jedna, ale nevadia oboje, takže oboje sú istota

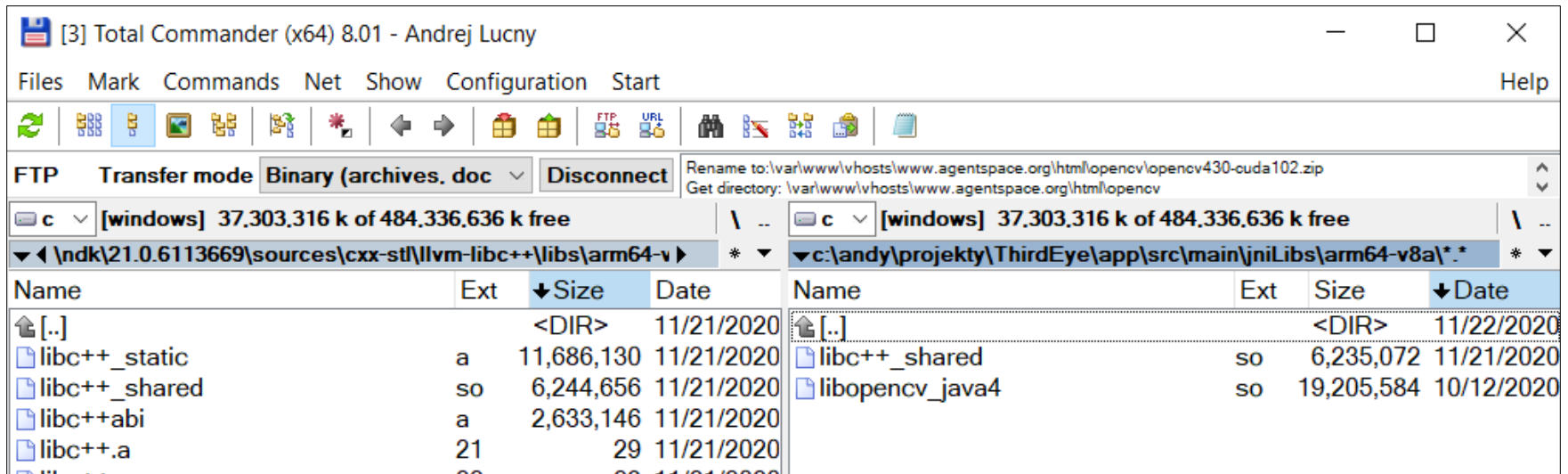


Tools /
 SDK Manager /
 Android SDK /
 tab SDK Tools /
 NDK (Side by side)
 OK

Nainštalujeme Native development kit (NDK).

Potrebujeme konkrétnu správnu verziu,
 pri našich verziách napr. 21.0.6113669

Ak verziu neutrafíme, pri buildovaní budeme vyzvaní nainštalovať tú
 správnu, takže na dva pokusy to dáme



skopírujeme knižnicu C++ `libc++_shared.so`

o ktorú sa opiera knižnice OpenCV z

`c:\Users\%USERNAME%\AppData\Local\Android\Sdk\ndk\21.0.6113669`

`\sources\cxx-stl\llvm-libc++\libs\arm64-v8a\`

do `app/src/main/jniLibs/arm64-v8a`

a `app/src/main/jniLibs/armeabi-v7a`

Name	Ext	Size	↓ Date
↑ [..]		<DIR>	11/21/2020
📄 pytorch_mobilenet	onnx	14,188,636	11/21/2020
📄 imagenet_classes	txt	22,675	11/21/2020

Do adresára [app/src/main/assets](#) nakopírujeme pomocné súbory: modely, názvy, ...

```
<FrameLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    xmlns:opencv="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">
```

```
    <org.opencv.android.JavaCameraView  
        android:id="@+id/CameraView"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:visibility="visible"  
        opencv:show_fps="false"  
        opencv:camera_id="any" />
```

```
</FrameLayout>
```

← Ako GUI použijeme
layout OpenCV, ktorý
zobrazuje obraz

nahradíme `app/src/main/res/layout/activity_main.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.learnopencv.tutorials.classification">
```

1/2

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.NoActionBar"> <!--Full screen mode-->
    <activity
        android:name=".MainActivity"
        android:screenOrientation="landscape"> <!--Screen orientation-->
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

<supports-screens android:resizeable="true"
    android:smallScreens="true"
    android:normalScreens="true"
    android:largeScreens="true"
    android:anyDensity="true" />
```

nahradíme [app/src/main/AndroidManifest.xml](#)

```
<!--Camera usage configuration-->
<uses-permission android:name="android.permission.CAMERA" />
```

2/2

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />
<uses-feature
    android:name="android.hardware.camera.autofocus"
    android:required="false" />
<uses-feature
    android:name="android.hardware.camera.front"
    android:required="false" />
<uses-feature
    android:name="android.hardware.camera.front.autofocus"
    android:required="false" />
```

← povolíme
← kameru

```
</manifest>
```

nahradíme [app/src/main/AndroidManifest.xml](#)

Kompilujeme cez Ctrl-F9

Keď pripojíme mobil cez usb k PC a vyberieme ho ako device, môžeme dať aj Run, čím sa aplikácia nainštaluje do mobilu a spustí.
(predtým ju treba odinštalovať)
a kým máme mobil pripojený vidíme aj pomocné výpisy

Istota ale je: (odinštalovať aplikáciu,) preniesť súbor [app-debug.apk](#) z [app/build/outputs/apk/debug](#) do mobilu do Download (Prevzaté súbory) a otvoriť ho cez Total Commander (apka do mobilu), čím ponúkne inštaláciu a po nainštalovaní ho ručne spustíme.

```
public class MainActivity extends CameraActivity
    implements CameraBridgeViewBase.CvCameraViewListener2 {

    private BaseLoaderCallback mLoaderCallback = new BaseLoaderCallback(this) {
        @Override
        public void onManagerConnected(int status) {
            switch (status) {
                case LoaderCallbackInterface.SUCCESS: {
                    Log.i(TAG, "OpenCV loaded successfully!");
                    mOpenCvCameraView.enableView();
                }
                break;
                default: {
                    super.onManagerConnected(status);
                }
                break;
            }
        }
    };
```

```
@Override
public void onResume() {
    super.onResume();
    // OpenCV manager initialization
    OpenCVLoader.initDebug();
    mLoaderCallback.onManagerConnected(LoaderCallbackInterface.SUCCESS);
}
```

inicializácia knižnice

inicializácia GUI


...
} Programujeme v [app\src\main\java\<domain>\<project>\MainActivity.java](#)

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    setContentView(R.layout.activity_main);

    // configure camera listener
    mOpenCvCameraView = (CameraBridgeViewBase) findViewById(R.id.CameraView);
    mOpenCvCameraView.setVisibility(CameraBridgeViewBase.VISIBLE);
    mOpenCvCameraView.setCvCameraViewListener(this);

    // further initialization
    // ...
}
```

**inicializácia preberania
obrázkov z kamery**



```
public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
    Mat frame = inputFrame.rgba();

    // modify frame by the OpenCV methods
    // ...

    return frame;
}
```

**spracovanie týchto
obrázkov**



Programujeme v `app\src\main\java\<domain>\<project>\MainActivity.java`

Pridáme komunikáciu s robotom cez USB:

využíva [com.felhr.usbserial](https://github.com/felHR85/UsbSerial) z

<https://github.com/felHR85/UsbSerial>

Táto knižnica sa dá vložiť do projektu vo forme jar-u,

t.j. do [app/libs/usbserial.jar](#)

Pričom nad ňou je vybudovaná nadstavba implementujúca ovládanie robota konkrétnymi príkazmi pre OpenBot,

takže túto preberme z OpenBot-a ako

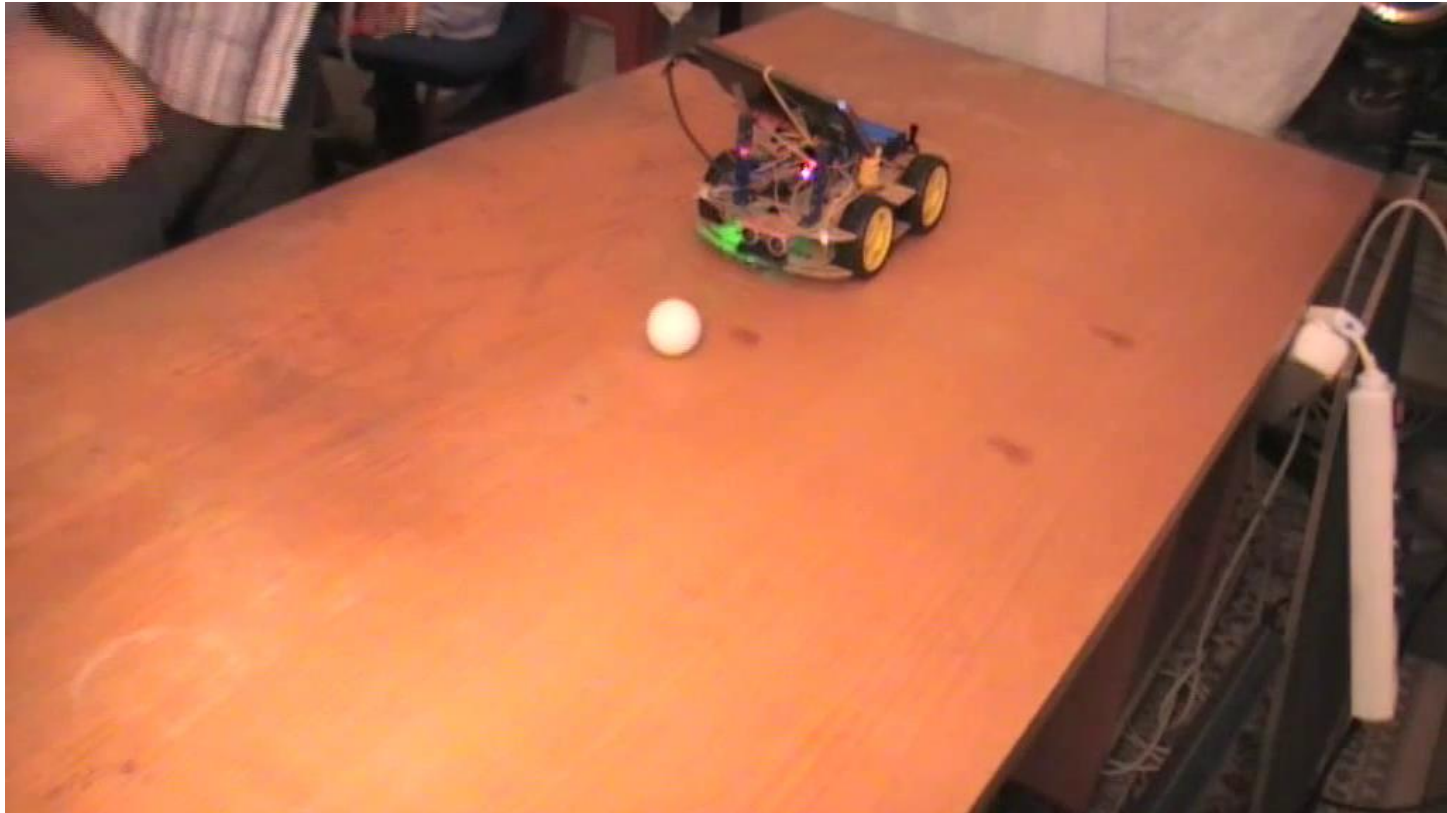
[app/src/main/java/org/openbot/env/UsbConnection.java](#)

Používanie USB na ovládanie robota nám žiaľ znemožní debugovanie z PC, takže programovanie sa stáva oveľa náročnejším na presnosť práce, aplikácia sa niekedy rúbe, keď ešte nemôže otvoriť USB port, atď.

Použitie modelu hlbokého učenia

- Na rozpoznávanie objektov používame `pytorch_mobilenet.onnx`
- MobileNet je architektúra hlbkej siete, nie príliš náročná na výpočtový výkon, určená je pre mobilné zariadenia
- Podporované formáty modelov sú v našom prípade tie ktoré podporuje OpenCV: `caffe`, `pb` a `onnx`
- Pri použití momentálne najobľúbenejšieho frameworku Pytorch teda natrénovaný model ukladáme ako `onnx`
- Do modelu vkladáme normalizovaný obraz a dostávame z neho vektor pravdepodobností príslušnosti k určitému objektu (klasifikátor)
- TODO: pridať je lokalizáciu objektov (detektor)
- Keď výpočet siete je náročný, zavádzame ho do softwaru ako servis

Demo



<https://youtu.be/VKqORhsmCDs>

<https://github.com/andyLucny/OpenBotWithOpenCV.git>

Ďakujem za pozornosť!

Seminár Robotika.SK

Mobilný telefón ako palubný počítač na robotovi

www.robotika.sk/seminar-archiv.php#prednaska42