

Projektni izvještaj



Nino Križanec, Rijeka 23.6.2017

Intel Edison: Android Things

Upoznavanje s Intel Edison Arduino pločicom

Intel Edison je razvojna pločica bazirana na Arduino platformi. Namijenjena je brzom prototyping-u i omogućava razvoj za razne platforme, podržava nativni arduino što je u projektu isprobano i radi besprijekorno. Također, osim arduino platforme podržava i linux za embedded sustave. Linux u obliku yocto distribucije nisam isprobao, ali je procedura instalacije jednostavna i prikazan je da radi pa ne bi trebalo biti problema s instalacijom. Nadalje za moj projekt korištena je po dogovoru najnovija inačica Androida u verziji „Android Things“.

Hardware

Sastavljanje pločice je poprilično jednostavno. U setu dolazi sve potrebno za assembly. Dovoljno je postaviti Intel Edison procesor u predviđeno mjesto, pričvrstiti ga maticama te pričvrstiti nožice na cijelu pločicu.

Za pokretanje pločice potreban je USB micro kabel. Također prilikom rada na projektu primjetio sam da pločica nekada ne želi raditi na određenim usb portovima i/ili kabelima. Primjetio sam da najbolje radi na USB 3.0 portovima. U literaturi se često spominje da pločica zahtjeva i externo napajanje. U mojem slučaju to nije bio problem.

Instalacija Firmware-a i Software-a

Za pokrenuti Intel Edison potrebno je flash-ati na njega početni firmware, tj bootloader. Ovaj dio mi je oduzeo neko vrijeme zbog problematičnosti sa driverima i softwareom. Potrebno je preuzeti Intel Flash Tool Lite koji će obaviti cijelu proceduru umjesto nas, naravno moguće je i ručno ali nema potrebe. [Intel Platform Flash Tool](#). Osim alata potrebno je instalirati i popratne drivere za Intel Edison i za Intel Android USB uređaje. Bez njih instalacija je nemoguća. [Intel Android USB Driver](#), [Intel Edison Drivers](#).

Nakon instalacije potrebnog software-a potrebno je preuzeti sistemski [image Android Things](#).

Ukoliko su instalirani svi driveri, pločicu se može spojiti usb mikro kabelom. Kabel se spaja na J16 port, a switch koji se nalazi iznad porta potrebno je namjestiti prema J16. Nakon spajanja u Intel Platform Flash Tool-u bi se trebao prikazati uređaj. Daljnja procedura je opisana na [web stranicama Android](#) i [web stranicama Intel-a](#).

Razvoj

Početak

Za početak razvoja instalirao sam Android Studio i popratne update i pluginove. Preuzeo sam osnovni projekt prilagođen Intel Edison-ua. Projekt je bez problema build-an i kompajliran u apk te je pokrenut na Intel Edisonu.

Nakon probavanja iskušao sam androidov support library za Input/Output. Naišao sam na probleme. Naime android je učestalo javljao kako su pinovi zauzeti i ne može ih alocirati. Ovo se ispravilo kada sam flash-ao najnoviju verziju Android Things-a.

Nadalje, kako je projekt trebao očitavati vrijednost senzora iz auta potrebno je bilo očitavati analogne vrijednosti. Ovo nije bilo moguće iz jednostavnog razloga što Android to još ne omogućuje. Došao sam do saznanja da je to u planu za budući razvoj, ali zbog strukture i slojeva apstrakcije na kojima radi Android nemoguće je napraviti vlastiti driver.

Kako je bilo nemoguće očitavati analogne vrijednosti okrenuo sam razvoj spajanju dva uređaja. Na oba uređaja instalirana je ista verzija Android Things. Početna je ideja bila spojiti uređaje putem Bluetooth Low Energy sustava. Iako je u teoriji poprilično jednostavno u praksi se pokazalo nemoguće. Velika limitacija je sam način na koji moderni Android sustav radi, naime učestalo od korisnika traži interakciju i potvrde za korištenje određenih resursa na uređaju. Uspio sam pokrenuti primjer BLE servera na jednom od uređaja, ali nažalost klijentska aplikacija je pisana za korištenje na običnom android uređaju i nije radila. Uspio sam povezati se na server koristeći 3rd party aplikaciju sa play trgovine zvanu [BLE scanner](#). Kako nigdje u literaturi nisam pronašao primjer kako spojiti dva android things uređaja BLE tehnologijom ako su oba dva bez ekrana i načina interakcije, tj čisto programski i preddefinirano odlučio sam probati spojiti uređaje običnim Bluetoothom.

Običan Bluetooth se također ispostavio kao neuspjeh. Naime kako bi se dva uređaja spojila potrebno je imati instanciran [BluetoothDevice](#) uređaj. Problem je u ovom slučaju to što se taj uređaj može instancirati samo od strane androida i to jedino prilikom očitavanja od prije uparenih uređaja ili prilikom pretrage za novim uređajima. Problem je nadalje bio to što bez ekrana i mogućnosti potvrde nisam uspio dobiti dovoljna prava za pristupiti informaciji o otkrivenom uređaju. Uređaj je točnije detektirao uređaj, a u aplikaciji postoji broadcast reciever metoda koja očekuje broadcast kada android sustav pronađe novi uređaj u blizini. Moja metoda nije imala privilegije pristupiti broadcastu iako je detektirala isti.

```
01:32:28.346 445-461/? W/BroadcastQueue: Permission Denial: receiving Intent {
act=android.bluetooth.device.action.FOUND flg=0x10 (has extras) } to ProcessRecord{cf9b26e
2457:com.example.androidthings.simplepio/u0a27} (pid=2457, uid=10027) requires
android.permission.ACCESS_COARSE_LOCATION due to sender com.android.bluetooth (uid 1002)
```

Kod 1 Greška u dopuštenjima

Iako greška izražava da ne aplikacija ne posjeduje dozvolu: „ACCESS_COARSE_LOCATION“ dozvola je dodjeljena, a problem je nedostatak interakcija bez ekrana koji bi omogućio potvrdu za dozvolu skeniranja bluetoothom.

Komunikacija

Nakon što nisam uspio povezati uređaje putem bluetooth-a probao sam ih spojiti putem wifi-a. To mi je pošlo za rukom, iako je proces kompliciraniji nego što je trebao biti zbog istih razloga kao i s bluetoothom.

Potrebno je uređaj spojiti na wireless mrežu koristeći se adb debug konzolom koja komunicira izravno na razini linux jezgre te pamti konfiguraciju za svako sljedeće paljenje uređaja.

Uređaji komuniciraju jednostavnim protokolom koji na web server postavlja varijablu, uređaj 1(server) mijenja varijablu dok uređaj 2(klijent) očitava varijablu. Za svoj sam primjer uzeo jednostavno blink-anje led diode, naime server zapisuje bool varijablu, koju klijent očitava i prema njoj pali ili gasi led diodu.

Daljnji problemi sa platformom su nastali pri svakom restartanju uređaja. Pri razvoju aplikacija u android manifest se zapisuju osnovni podatci o aplikaciji kao i intenti na koje ona odgovara. U Android Things-u postoji intent koji se oglašava pri paljenju uređaja. U mojem slučaju on jednostavno nikad nije proradio već sam morao manualno pokrenuti aplikaciju nakon svakog paljenja.

```
<!-- Launch activity automatically on boot -->
<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.IOT_LAUNCHER"/>
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

Kod 2 Intent filter za pokretanje

Zaključak

Iako je Android Things tek u začecima mislim da bi mogao zauzeti velik dio embedded tržišta zbog jednostavnosti povezivanja sa novim tehnologijama. Također Android Things već ima i svoj google play market što će omogućiti krajnjim korisnicima da ne moraju sami programirati svoje aplikacije već da preuzmu gotove aplikacije za npr smart house ili nešto drugo.

Nadam se da će Android Development tim nastaviti raditi na ovoj platformi i da će pojednostaviti spajanje uređaja i dozvoliti veću kontrolu korištenja pinova.

Reference

<https://01.org/android-ia/downloads/intel-platform-flash-tool-lite>

<https://software.intel.com/android/articles/intel-usb-driver-for-android-devices>

<http://downloadmirror.intel.com/24909/eng/IntelEdisonDriverSetup1.2.1.exe>

<https://software.intel.com/en-us/articles/installing-android-things-on-intel-edison-kit-for-arduino>

<https://play.google.com/store/apps/details?id=com.macdom.ble.blescanner>

<https://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>