Forgia Paolo

SSSE  SIG

Manuale operatore

Indice

[1 Setup ambiente di sviluppo 2](#_Toc122702161)

[1.1 Visual Studio 2](#_Toc122702162)

[1.2 Intel Hardware Accelerated Execution Manager (HAXM) 2](#_Toc122702163)

[2 Configurazione 2](#_Toc122702164)

[2.1 Emulatore 2](#_Toc122702165)

[2.2 Telefono 3](#_Toc122702166)

[2.3 Compilare 3](#_Toc122702167)

[3 Versioning 4](#_Toc122702168)

[3.1 Git 4](#_Toc122702169)

# Setup ambiente di sviluppo

## Visual Studio

Sono disponibili varie versioni di Visual Studio, qualsiasi di queste va bene, consigliata la versione Community siccome è gratuita.

Durante l’installazione selezionare l’opzione “.NET Multi-platform App UI development)

Graphical user interface, text, application

Description automatically generated

Link per il download: <https://visualstudio.microsoft.com/downloads/>

## Intel Hardware Accelerated Execution Manager (HAXM)

Se si vuole approfittare dell’emulatore sul pc e si possiede un processore Intel, bisogna scaricare HAXM, il quale migliora notevolmente le performance dell’emulazione.

Se invece si vuole compilare il codice direttamente su un reale dispositivo non è necessaria l’installazione di HAXM.

Link per il download: <https://github.com/intel/haxm>

# Configurazione

## Emulatore

Nel caso si volesse far partire l’applicativo su un emulatore è necessario crearne uno.

Text

Description automatically generated

È possibile crearne uno specificando le proprietà hardware e versioni dell’OS[[1]](#footnote-1), oppure partire da un preset esistente, suggerisco la seconda.

Graphical user interface

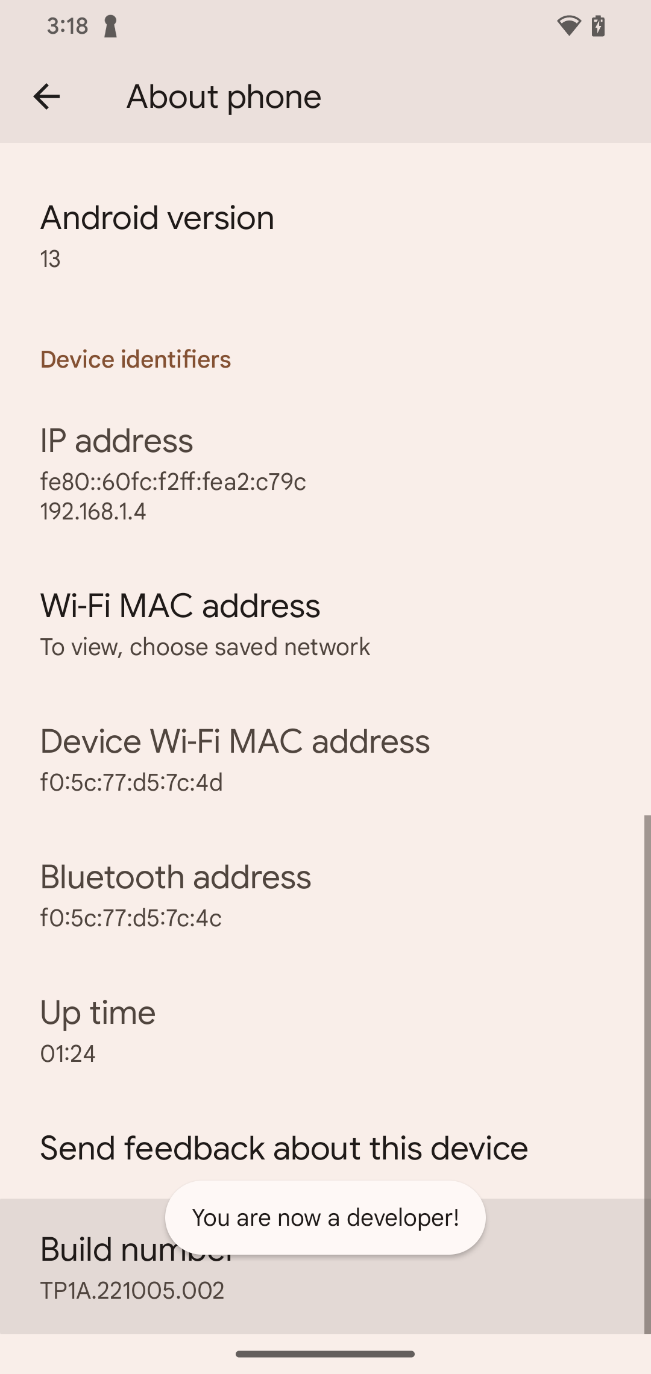
Description automatically generated

## Telefono

Per compilare il codice direttamente sul telefono, bisogna innanzitutto attivare la modalità sviluppatore sul telefono.

Per attivare la modalità sviluppatore andare nelle impostazioni, sistema, informazioni sul telefono, e premere ripetutamente su “Numero build” fintanto che la modalità sviluppatore viene attivata.

Dopodiché sotto sistema, sarà apparsa la voce “Opzioni sviluppatore”, all’interno attivare “Debug USB”. Questo permetterà di compilare il codice direttamente sul telefono.



## Compilare

Selezionare il dispositivo su cui si vuole lanciare la compilazione e lanciare la compilazione.

Text

Description automatically generated

# Versioning

## Git

Per il verisoning è stato usato Git, è necessario però abilitare il supporto per i file con il nome oltre i 260 caratteri. Per il nome si intende il percorso delle cartelle e il nome del file.

Il comando è il seguente:

git config --system core.longpaths true

Link per il download: <https://git-scm.com/downloads>

## Scaricare progetto

Per scaricare il progetto lanciare il seguente comando, nella cartella in cui si vuole avere il progetto:

git checkout <https://github.com/PaoloForgia/CPTRoverControl.git>

# Compatibilità

## Android

Il progetto è stato compilato con il supporto dalla versione Android 4.4, così da andare a supportare il 99.6% dei dispositivi in commercio[[2]](#footnote-2).

A picture containing chart

Description automatically generated

Il target per lo sviluppo è stato impostata l’ultima versione di Android come suggerito nella documentazione di Xamarin[[3]](#footnote-3).

Graphical user interface, text, application

Description automatically generated

## Librerie

|  |  |  |  |
| --- | --- | --- | --- |
| Libreria | Versione utilizzata | Versione minima | Scopo |
| Xamarin.Forms | 5.0.0.2545 | 5.0.0.0 | Framework per sviluppo mobile |
| Xamarin.Essentials | 1.7.3 | 1.7.0 | Utility |
| Xamarin.CommunityToolkit | 2.0.5 | 2.0.0 | Utility |
| Plugin.BluetoothClassic | 1.1.5 | 1.0.0 | Connessione Bluetooth |
| sqlite-net-pcl | 1.8.116 | 1.8.0 | Connessione database |
| Microsoft.NET.Test.Sdk | 17.4.0 | 17.0.0 | Testing |
| NUnit | 3.13.3 | 3.0.0 | Testing |
| NUnit.Analyzers | 3.3.0 | 3.0.0 | Testing |
| NUnit3TestAdapter | 4.2.1 | 4.0.0 | Testing |

# Deploy app

Per distribuire quanto sviluppato su un telefono bisogna generare una nuova versione dell’applicazione sottoforma di app in formato Android (file con estensione .apk).

## Android manifest

Prima di creare una nuova release, potrebbe essere necessario correggere alcune proprietà, quali versione dell’applicazione, target o permessi.

Per fare ciò aprire il progetto Android (RoverControlApp.Android), tasto destro su Properties e selezionare Properties nel menu.

Graphical user interface, application

Description automatically generated

## Cambiare tipo build

Nel menu selezionare Build e in seguito “Configuration Manager…”.

Graphical user interface, text, application

Description automatically generated

Impostare la configurazione per “RoverControlApp.Android” su Release invece che Debug.

Ricompilare il progetto selezionando Build e poi Rebuild Solution.

Graphical user interface, text, application

Description automatically generated

## Generare il file apk

Tasto destro sul progetto “RoverControlApp.Android” e selezionare “Archive…”. Questo farà aprire l’Archive Manager, tramite il quale si potrà distribuire l’app.

Verrà creato un elemento come il seguente.

Graphical user interface, text, application, email

Description automatically generated

Selezionarlo e cliccare su “Distribute…”.

La modalità di distribuzione “Ad Hoc”, importare la identity precedentemente creata, inserire la password.

Nel caso si volesse creare una nuova Signing Identity selezionare il bottone con il più e crearne una nuova.

In seguito, selezionare “Save As” e la cartella di destinazione.

# Installazione App

## Scaricare l’app

Siccome l’applicazione non è disponibile sullo store bisognerà passarsela tramite una piattaforma di file hosting (Dropbox, Google Drive, ecc.).

## Abilitare installazione di app sconosciute

Passarsi l’applicazione nel modo più pratico e salvarla.

Per installare applicazioni non provenienti dal Google Play Store abilitare un’opzione nelle impostazioni di sistema. Sotto impostazioni, app, accesso speciale per le app, installa app sconosciute; ora bisogna dare l’autorizzazione a “Files by Google” o un altro file manager installato sul dispositivo.

## Installazione

Nel caso si volesse installare l’app su un dispositivo che si è precedentemente utilizzato per il debug è consigliato disinstallare l’applicazione precedentemente generata.

Una volta autorizzato aprire il file manager e cliccare sull’applicazione precedentemente scaricata e fare partire l’installazione.

Graphical user interface, text, application

Description automatically generated

## Installazione fallita

### Aggiornamento di una versione esistente

Se una versione precedente era già installata sul dispositivo, è possibile che installare una nuova versione possa dare problemi.

Se dovesse succedere disinstallare l’applicazione e riprovare.

### Altro

Provare a ri-compilare nuovamente l’applicazione nel formato apk e riprovare, questo potrebbe risolvere il problema.

Se dovesse fallire nuovamente, eseguire il deploy in modalità debug (come in fase di sviluppo) e investigare la causa.

1. Sistema Operativo [↑](#footnote-ref-1)
2. https://9to5google.com/2022/05/20/android-2022-distribution-numbers-chart/ [↑](#footnote-ref-2)
3. https://learn.microsoft.com/en-us/xamarin/android/app-fundamentals/android-api-levels?tabs=windows#target-android-version [↑](#footnote-ref-3)