

Bot Telegram su Raspberry

Lo scopo di questa attività è integrare il sensore e l'attuatore nell'ambiente Telegram. Nel seguito si vedrà come realizzare un semplice BOT Telegram, che poi potrà essere arricchito con le funzionalità di controllo del sensore e dell'attuatore.

*Nella **Sezione A** si configura il BOT utilizzando l'applicazione Telegram su cellulare o PC; nella **Sezione B** è riportato un esempio di programma che realizza un BOT; nella **Sezione C** si descrive il test del BOT*

Sezione A: configurazione BOT

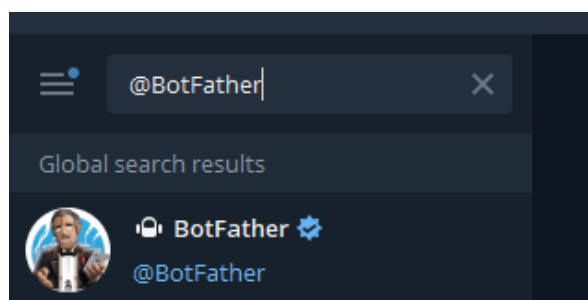
Parte 1: installazione Telegram

Fase 1.1: installare Telegram sul cellulare

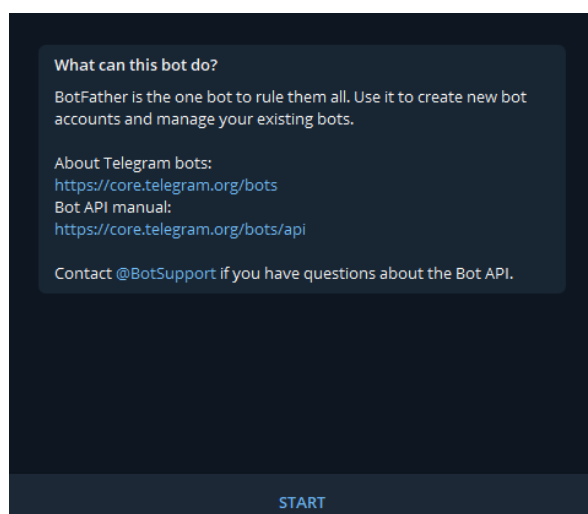
Fase 1.2: opzionalmente, installare Telegram su pc e associarlo all'applicazione installata su cellulare

Parte 2: registrazione del BOT

Fase 2.1: dall'applicazione Telegram, cercare **@BotFather** e selezionarlo per inviargli messaggi



Fase 2.2: cliccare su **START**



Fase 2.3: cliccare su [/newbot](#)

```
BotFather
bot

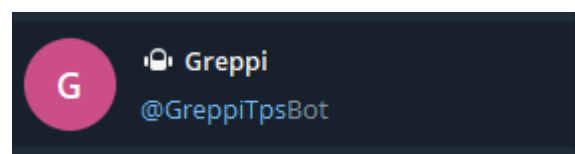
/newbot - create a new bot
/mybots - edit your bots [beta]

Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games [beta]
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletgame - delete an existing game
```

Fase 2.4: assegnare un nome (ad esempio *Greppi*) ed uno username (ad esempio *GreppiTpsBot*): compariranno entrambi quando si interagirà col BOT. Conservare poi il token che viene assegnato al BOT e che verrà utilizzato nel programma.



Sezione B: programma di esempio

Parte 3: installazione librerie

Fase 3.1: installare *python-telegram-bot*

```
python3 -m pip install python-telegram-bot
```

Parte 4: esempio telebot.py

Fase 4.1: sostituire il TOKEN nel programma di esempio (riportato in coda in versione copiabile)

Il programma accetta due comandi, oltre a */start* e */help*: */saluta* e */insulta*; per entrambi restituisce un semplice messaggio. I due comandi sono anche lanciabili tramite due bottoni.

```
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackQueryHandler, CallbackContext
from telegram import InlineKeyboardButton, InlineKeyboardMarkup, Update, KeyboardButton, ReplyKeyboardMarkup

#-----
# costanti
#-----
TOKEN="5075804894:AAFmIxez9lHSBtqUBstflMgurtDKKn2_vgc"
PROXY={'proxy_url': 'http://proxy:3128'}
#-----
# funzioni per la gestione dei diversi messaggi e comandi
#-----
# messaggio /start
def start(aggiornamento, contesto):
    help(aggiornamento, contesto)

# messaggio /help
def help(aggiornamento, contesto):
    tastiera = [
        [
            KeyboardButton("/saluta: manda un saluto"),
            KeyboardButton("/insulta: manda un insulto"),
        ]
    ]
    tastiera_markup = ReplyKeyboardMarkup(tastiera)
    aggiornamento.message.reply_text('Scegli una funzione:', reply_markup=tastiera_markup)

# messaggio senza /
def echo(aggiornamento, contesto):
    help(aggiornamento, contesto)

# messaggio /saluta
def saluta(aggiornamento, contesto):
    aggiornamento.message.reply_text('Ciao!')

# messaggio /insulta
def insulta(aggiornamento, contesto):
    aggiornamento.message.reply_text('Babbo!')

# errore
def error(aggiornamento, contesto):
    print(contesto.error)

#-----
# connessioni Telegram
#-----

# crea l'updater telegram passandogli la chiave ottenuta da Telegram in fase di registrazione del Bot
# senza Proxy
updater = Updater(TOKEN)
# con il Proxy
#updater = Updater(TOKEN, request_kwargs=PROXY)

# dispatcher per la registrazione delle funzioni di gestione messaggi telegram
dp = updater.dispatcher

# registrazione delle funzioni per la gestione messaggi telegram
dp.add_handler(CommandHandler("start", start))
dp.add_handler(CommandHandler("help", help))
dp.add_handler(CommandHandler("saluta", saluta))
dp.add_handler(CommandHandler("insulta", insulta))
dp.add_handler(MessageHandler(Filters.text, echo))
dp.add_error_handler(error)

# avvia l'attesa di messaggi telegram e attende
updater.start_polling()
updater.idle()
```

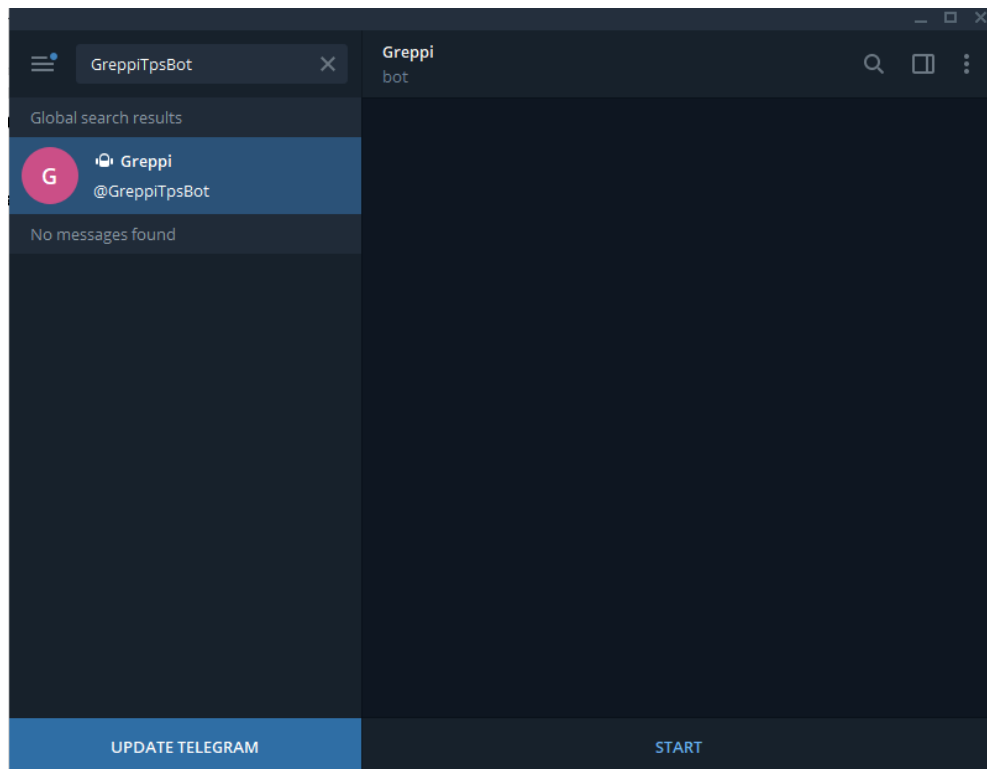
Fase 4.2: lanciare telebot.py

```
python3 telebot.py
```

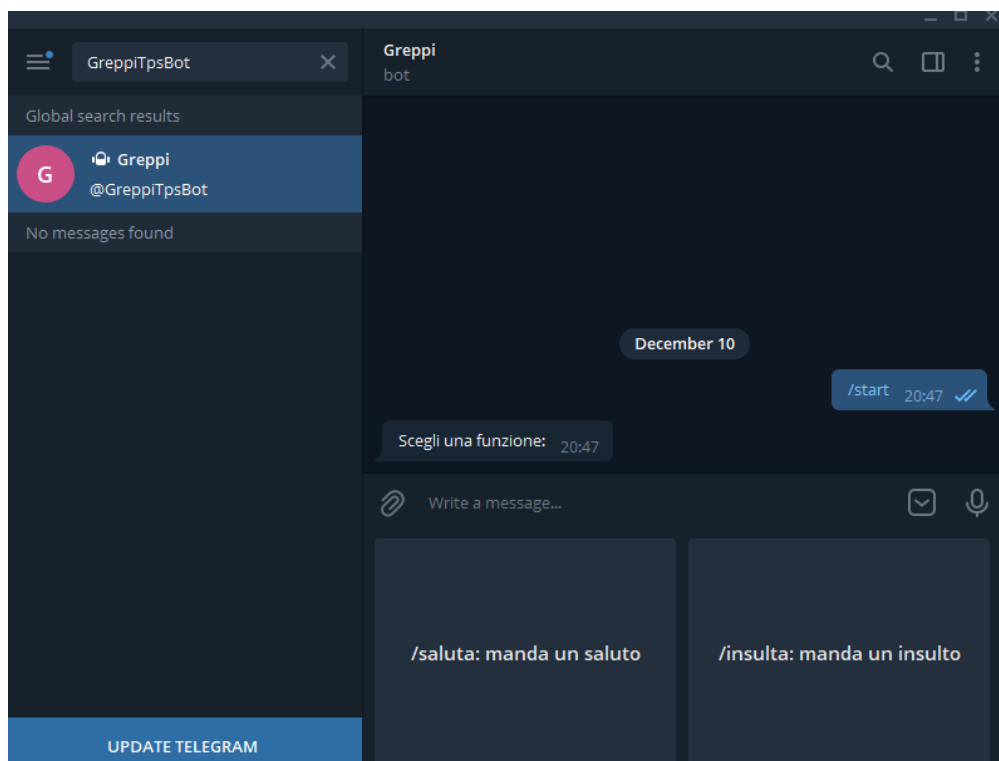
Sezione C: test del BOT

Parte 5: test del bot

Fase 4.1: da Telegram cercare il BOT



Fase 4.2: selezionarlo e cliccare *START*



```

from telegram.ext import Updater, CommandHandler, MessageHandler, Filters,
CallbackQueryHandler, CallbackContext
from telegram import InlineKeyboardButton, InlineKeyboardMarkup, Update, KeyboardButton,
ReplyKeyboardMarkup
#-----
# costanti
#-----
TOKEN="5075804894:AAFmIxez9lHSBtqUBztflMgurtDKKn2_vgo"
PROXY={'proxy_url': 'http://proxy:3128'}
#-----
# funzioni per la gestione dei diversi messaggi e comandi
#-----
# messaggio /start
def start(aggiornamento, contesto):
    help(aggiornamento, contesto)

# messaggio /help
def help(aggiornamento, contesto):
    tastiera = [
        [
            KeyboardButton("/saluta: manda un saluto"),
            KeyboardButton("/insulta: manda un insulto"),
        ]
    ]
    tastiera_markup = ReplyKeyboardMarkup(tastiera)
    aggiornamento.message.reply_text('Scegli una funzione:', reply_markup=tastiera_markup)

# messaggio senza /
def echo(aggiornamento, contesto):
    help(aggiornamento, contesto)

# messaggio /saluta
def saluta(aggiornamento, contesto):
    aggiornamento.message.reply_text('Ciao!')

# messaggio /insulta
def insulta(aggiornamento, contesto):
    aggiornamento.message.reply_text('Babbeo!')

# errore
def error(aggiornamento, contesto):
    print(contesto.error)

#-----
# connessioni Telegram
#-----

# crea l'updater telegram passandogli la chiave ottenuta da Telegram in fase di registrazione
del Bot
# senza Proxy
updater = Updater(TOKEN)
# con il Proxy
#updater = Updater(TOKEN, request_kwargs=PROXY)

# dispatcher per la registrazione delle funzioni di gestione messaggi telegram
dp = updater.dispatcher

# registrazione delle funzioni per la gestione messaggi telegram
dp.add_handler(CommandHandler("start", start))
dp.add_handler(CommandHandler("help", help))
dp.add_handler(CommandHandler('saluta', saluta))
dp.add_handler(CommandHandler('insulta', insulta))
dp.add_handler(MessageHandler(Filters.text, echo))
dp.add_error_handler(error)

# avvia l'attesa di messaggi telegram e attende
updater.start_polling()
updater.idle()

```