Protocollo di comunicazione con il robot Spykee (incompleto...)

1. Connessione
Sono disponibili varie modalità di connessione tra il robot e il computer. La più semplice (e l'unica utilizzata da noi...) è l'utilizzo di una rete Wi-Fi ad hoc che viene creata dal robot (il robot funge anche da server DHCP, quindi non è necessario impostare manualmente i parametri di configurazione dell'interfaccia di rete), e il cui SSID è `SPYKEE` seguito da un codice identificativo. I comandi del protocollo che servono a impostare la rete in maniera differente non sono stati testati, e non sono completamente specificati nel seguito.

Discovery
Per ricavare l'indirizzo IP del robot, è necessario inviare un pacchetto via UDP in broadcast sulla rete di Spykee (indirizzo di broadcast `172.17.6.255`, porta `9000)`. Il pacchetto di richiesta, di lunghezza 5 byte, è il seguente: `{ 68, 83, 67, 86, 01 }`, ossia "`DSCV`" seguito dal byte 1 (`DISCOVER_PACKET_TYPE_REQUEST`).
Il robot risponde inviando via UDP un pacchetto dal suo indirizzo IP col seguente formato:

- Header (6 byte): "`DSCV`" seguito dal byte 2 (`DISCOVER_PACKET_TYPE_REPLY`), seguito da un byte che contiene la lunghezza del payload
- Payload (variabile): è testuale e contiene i seguenti campi, separati da virgola:
  - `uid=`<codice identificativo del robot>
  - `connected-user=`<utente attualmente connesso>. Se nessun utente è connesso, il campo è omesso

Autenticazione
Una volta noto l'indirizzo IP del robot, è possibile iniziare una connessione TCP sulla porta 9000. Per autenticarsi, bisogna mandare un messaggio di tipologia `PACKET_TYPE_AUTH_REQUEST`. Il payload del messaggio deve essere formato da un byte che contiene la lunghezza del nome utente, il nome utente (di default '`admin`'), seguito dalla lunghezza della password e dalla password (di default '`admin`').

Formato dei messaggi
Tutti i messaggi inviati o ricevuti da Spykee attraverso la connessione TCP sono composti da un header di formato comune e da un payload, specifico del messaggio. L'header è costituito da 5 byte il cui significato è riportato di seguito.

'P'	'K'	TYPE	LENGTH

Per 'P' e 'K' si intendono i *caratteri* 'P' e 'K' in formato ASCII (80 e 75), e LENGTH è la dimensione in byte del payload. I valori possibili per il campo TYPE sono i seguenti:

```
PACKET_TYPE_NONE              0
PACKET_TYPE_AUDIO             1
PACKET_TYPE_VIDEO             2
PACKET_TYPE_POWER             3
PACKET_TYPE_LED               4
PACKET_TYPE_MOVE              5
PACKET_TYPE_FILE              6
PACKET_TYPE_PLAY              7
PACKET_TYPE_STOP              8
PACKET_TYPE_VOLUME            9
PACKET_TYPE_AUTH_REQUEST     10
PACKET_TYPE_AUTH_REPLY       11
PACKET_TYPE_MUTE             12
PACKET_TYPE_CONFIG           13
PACKET_TYPE_WIRELESS_NETWORKS 14
PACKET_TYPE_STREAMCTL        15
PACKET_TYPE_ENGINE           16
PACKET_TYPE_LOG              17
PACKET_TYPE_MOTOR_SPEED      18
```

2. Messaggi inviati dal robot

Power level (tipo: `PACKET_TYPE_POWER`)
Indica il livello di carica della batteria, espresso in percentuale da 0 a 100 e contenuto nell'unico byte del payload. Quando il robot è in carica, viene inviato un livello di carica negativo.

Video (tipo: `PACKET_TYPE_VIDEO`)
Il payload contiene un frame catturato dalla telecamera e compresso in formato JPEG, risoluzione 320x240

"Event Music closed" (tipo: `PACKET_TYPE_STOP`)
Il payload è vuoto. (non mi è chiaro il significato!)

Audio (tipo: `PACKET_TYPE_AUDIO`)
Il payload contiene i dati del buffer dell'audio registrato dal microfono onboard di Spykee (raw 16-bit signed samples, 16000HZ)

Messaggi di controllo (tipo: `PACKET_TYPE_ENGINE`)
Payload di lunghezza 1 byte, che rappresenta il messaggio. Alcuni dei payload sono:

- **1**: segnala che il robot è rimosso dalla stazione di carica (`MESSAGE_TYPE_ACTIVATE`)
- **2**: segnala che il robot è agganciato alla stazione di carica (`MESSAGE_TYPE_DEACTIVATE`)
- **3**: segnala che la batteria è carica (`MESSAGE_TYPE_BATTERY_CHARGED`)
- **8**: segnala che la base non è stata trovata (`MESSAGE_TYPE_BASE_NOT_FOUND`)

Nota: non sono sicuro del significato di questi messaggi!

3. Messaggi ricevuti dal robot

Controllo stream audio e video (tipo: `PACKET_TYPE_STREAMCTL`)
Il payload è formato da due byte. Il primo contiene l'ID dello stream da comandare e il secondo contiene lo stato a cui porre lo stream (`0` = spento, `1` = acceso). ID riconosciuti:

- **1**: `STREAM_ID_VIDEO` (avvia o ferma l'acquisizione del video)
- **2**: `STREAM_ID_AUDIO_IN` (avvia o ferma cattura audio da SpyKee)
- **3**: `STREAM_ID_AUDIO_OUT` (avvia o ferma l'altoparlante di SpyKee. Non so questo cosa riproduce, probabilmente l'audio ricevuto dal computer, PACKET_TYPE_AUDIO, oppure...)

Led (tipo: `PACKET_TYPE_LED`)
Comanda i led montati sul robot. Il payload è formato da due byte. Il primo costituisce l'ID del led da comandare, e il secondo è lo stato di attivazione (0 = spento, 1 = acceso). ID riconosciuti:

- **0**: `LED_USER_FLASH`
- **1**: `LED_USER_LEFT`
- **2**: `LED_USER_RIGHT`

Motori (tipo: `PACKET_TYPE_MOVE`)
Comandi per muovere i cingoli di Spykee. Il primo bit del payload è il comando al cingolo sinistro, il secondo comanda il cingolo destro. Entrambi i bit contengono un intero da -90 a +90

Engine (tipo: `PACKET_TYPE_ENGINE`)
Può essere uno dei sottotipi tra:

- **5:** `MESSAGE_TYPE_CHARGE_STOP`: (undock: sgancia il robot dalla stazione di carica)
- **6:** `MESSAGE_TYPE_BASE_FIND:` (dock: cerca di agganciarsi alla base)
- **7:** `MESSAGE_TYPE_BASE_FIND_CANCEL:` (cancel dock: annulla la ricerca della base)

Audio (tipo: `PACKET_TYPE_AUDIO`)
Il payload contiene i dati del buffer dell'audio in formato compresso (quale?) probabilmente viene riprodotto attraverso gli altoparlanti di Spykee se è attivato STREAM_ID_AUDIO_OUT

Suoni (tipo: `PACKET_TYPE_PLAY`)
Riproduce un suono. Il primo (e unico) bit del payload è l'ID del suono da riprodurre:

- **0**: `FILE_ID_SOUND_EVENT_0` (*alarm.mp3*)
- **1**: `FILE_ID_SOUND_EVENT_1` (*bomb.mp3*)
- **2**: `FILE_ID_SOUND_EVENT_2` (*lazer.mp3*)
- **3**: `FILE_ID_SOUND_EVENT_3` (*ah-ah-ah.mp3*)
- **4**: `FILE_ID_SOUND_EVENT_4` (*engine.mp3*)
- **5**: `FILE_ID_SOUND_EVENT_5` (*robot.mp3*)
- **6**: `FILE_ID_SOUND_EVENT_6` (*user0.mp3*)
- **7**: `FILE_ID_SOUND_EVENT_7` (*user1.mp3*)
- **16**: `FILE_ID_SOUND_EVENT_SNAPSHOT` (*camera.mp3*)
- **17**: `FILE_ID_SOUND_EVENT_RECORD` (*bip.mp3*)
- **18**: `FILE_ID_SOUND_EVENT_BIPBIP`
- **19**: `FILE_ID_SOUND_EVENT_BIPBIPBIP`
- **20**: `FILE_ID_SOUND_EVENT_BIPBIPBIPBIP`
- **64**: `FILE_ID_MUSIC`

Alcuni suoni cambiano anche lo stato di alcuni led. `FILE_ID_SOUND_EVENT_RECORD` setta `isRecording` a 1 oppure a 0, ma non so che effetti abbia.

Volume (tipo: `PACKET_TYPE_VOLUME`)
Volume degli altoparlanti di SpyKee, rappresentato dall'unico byte del payload come percentuale.

Stop (tipo: `PACKET_TYPE_STOP`)
Ferma la musica correntemente in riproduzione

Invio di file (tipo: `PACKET_TYPE_FILE`)
Viene utilizzato per inviare un file al robot: è possibile caricare un file audio da associare a uno degli ID impostabili dall'utente, una nuova versione del firmware, oppure un nuovo file di configurazione. Da capire bene il formato del pacchetto (viene gestito da `Server_Msg_File` in `robot.c)` e del file di configurazione: per ora comunque non serve.

Configurazione (tipo: `PACKET_TYPE_CONFIG`)
Se il payload è vuoto, il robot risponde inviando una serie di messaggi contenenti il file di configurazione corrente. Altrimenti, inviando un opportuno payload, è possibile modificare gli account (username e password), le impostazioni di connessione alla rete wifi (SSID, canale, ...), e altre impostazioni relative al robot. Le impostazioni verranno salvate nella NVRAM (memoria non volatile). Non ho indagato sul formato del pacchetto e su tutte le impostazioni accettate, comunque è gestito dalla funzione `Server_Msg_Config` in `robot.c` L'utente correntemente autenticato deve avere privilegi di amministrazione.

Log (tipo: `PACKET_TYPE_LOG`)
Servono privilegi di admin, gestito da `Server_Msg_Log` in `robot.c.` Il robot risponde con un pacchetto di tipo `PACKET_TYPE_LOG` che contiene il file di log.

Motor speed (tipo: `PACKET_TYPE_MOTOR_SPEED`)
Salva il payload su NVRAM alla voce 'motor-speed':

```
NVRamSet("motor-speed", NetBuffer);
NVRamCommit();
```

(netbuffer è il payload). Non so quale sia il significato.

Scansione reti wireless (tipo: `PACKET_TYPE_WIRELESS_NETWORKS`)
Servono privilegi di amministrazione. Non serve, viene utilizzato per connettere il robot ad altre reti wifi, in particolare per restituire un elenco di reti wifi rilevate dal robot. Se serve, nel codice del firmware viene gestito dalla procedura `Server_Msg_WirelessDiscover()` in `robot.c`. Per ora non serve. Invia un messaggio del tipo `PACKET_TYPE_WIRELESS_NETWORKS` con i risultati della ricerca