SilverVISE

Simple Android telemetry application for tracking data from quadcopters flashed with silverxxx firmware modified to work with BLE technology. Made for pilots that has no Devo or other telemetry compatible controllers supported by silverxxx firmware or want to track some more telemetry data and check PID values.

Application can track and warn you about bind status, power-up and flying time, battery level (voltage, percentage shown in two different ways etc.), RC signal level and bluetooth RSSI, current flight mode (acro/level), speed (normal/high), acro PIDs etc., all with audio, vibration and visual alarms with configurable treshold values and other options.

Quick info:

Here is a condensed summary about the "Silvervise" app and it's uses

- Silvervise uses the Bluetooth-low energy BTLE communication standard
- App works on android 4.4 and higher
- Display of battery , pids, and other info on the phone
- A specific "App" protocol needs to be set on the quad
- One way communication is possible only
- Some phones have issues due to nonstandard implementation and Android bugs
- App is made by RCGroups user SilverAG

Latest stable, official application version download link and thread for reporting bugs, comments, questions, critics etc.:
https://www.rcgroups.com/forums/showthread.php?2759160-SilverVISE-Android-application-for-tracking-TLM-data-from-silverised-quads

Latest firmwares with SilverVISE support:

- **For H101 and H8S boards:** https://github.com/silver13/H101-acro
- **For H8 green board:** https://github.com/silver13/h8mini-dual
- **For H8 blue board:** https://github.com/silver13/H8mini_blue_board
- **For BWHOOP B-03 board**: https://github.com/silver13/BoldClash-BWHOOP-B-03 (official) and https://github.com/silverag123/BoldClash-BWHOOP-B-03 (unofficial - supports dual telemetry and switching between SilverVISE and Devo telemetry with PIDs)

Special thanks to all who tried and tested application, and very big thanks to users from RC GROUPS forum who helped to develop this (listed in random order): silverxxx, MegaloRex, bikemikem, Bonaventura, SirDomsen, sammy4703, samhain616, nigelsheffield, ltwvince, Nitro123, lostmyquad, Ian444, ArbitraryNotion, Markus Gritsch, 0a23cb7af4d242, eitama, Appelmoesman and others I forget to mention (sorry :( ).

Configuration needed to work:

- Android device (phone, tablet...) with Android version 4.4.2 or later with standard bluetooth suport and with BLE (Bluetooth Low Energy) support and installed SilverVISE application
- Quadcopter with supported board flashed with silverxxx firmware with added support for this application

How to prepare firmware and quadcopter to enable transmitting telemetry data:

FIRMWARE CHANGES ARE MADE ONLY FOR SPECIFIED FIRMWARES FOR NOW! DO NOT TRY TO USE THESE FILES WITH OTHER QUADCOPTERS/BOARDS AND DON'T USE PARTS OF THIS CODE IN OTHER FIRMWARES BECAUSE YOU MAY DAMAGE YOUR QUADCOPTERS! ALSO, PLEASE READ ALL DETAILS, INSTRUCTIONS AND DESCRIPTIONS WRITTEN INSIDE COMMENTS IN RX_BAYANG_BLE_APP.C REGARDING ALL OPTIONS FOR SILVERVISE TELEMETRY!

In `config.h`:

Turn on SilverVISE support (RX_BAYANG_BLE_APP) and turn off Devo and Eddystone beacon support (RX_BAYANG_TELEMETRY and RX_BAYANG_BLE or RX_BAYANG_PROTOCOL_TELEMETRY and RX_BAYANG_PROTOCOL_BLE_BEACON - depending on a firmware you use):

```
//rx protocol selection
//#define RX_BAYANG_TELEMETRY
//#define RX_BAYANG_BLE
#define RX_BAYANG_BLE_APP
```

or:

```
//#define RX_BAYANG_PROTOCOL_TELEMETRY
//#define RX_BAYANG_PROTOCOL_BLE_BEACON
#define RX_BAYANG_BLE_APP
```

In `rx_bayang_ble_app.c`

- Name your quad - MY_QUAD_NAME **"XXXXXX"** - for telemetry with SilverVISE application, you must define unique name of each quad you flash. Names between quadcopters must not be the same if you plan to fly more than one in the same time. Name must be entered inside quotation marks (here represented with XXXXXX. Be original and keep up to 6 characters (shorter than 6 is OK but any longer name will be cut to first 6 characters). You can use letters A to Z, numbers 0 to 9 and some special characters like # / - _ etc. Please use only caps lock letters because name is not case sensitive. Avoid using blanks anywhere in name!
- Set quad ID in MY_QUAD_ID declaration - MY_QUAD_ID defines unique ID for MAC address. Keep or replace default value 123 with any value between 1 and 255 (use ONLY values between 1 and 255). **PLEASE READ DETAILS INSIDE COMMENTS IN RX_BAYANG_BLE_APP.C REGARDING THIS!**
- Set other SilverVISE related options if needed...
- Build files and flash your quad

In case you have problems with receiving data (some H101 boards have signal problems with some Android devices especially when you hit full throttle), check application settings screen and

"Telemetry scanning/data receiving mode" option, and also check `rx_bayang_ble_app.c` for additional options that might help you to get stable connection (explanations what is what is inside `rx_bayang_ble_app.c`. Also, read section "<u>Possible problems you might get using app on your Android device and additional steps to check and solve</u> "...

Setup phone to receive and track TLM data:

Download ZIP with APK file, unpack APK from ZIP and install application (unpacked APK file) on your Android device - here you might have problems because application is not downloaded from Google PlayStore so your Android device might reject installation. In that case, you need to allow installation of APK files from unknown sources. Here is how to do that: go to "Settings", then "Security" and there check "Unknown sources" (I hope menus looks like that on phones set on English because all my test phones are in Croatian so I'm guessing 🙂). Then download file and tap/click it. If it won't install, go to "file manager" and under "applications" or "downloads" (or where you unpacked APK file) find `SilverVISE-v2_00.apk` file and then click/tap on it and it will allow you to install. When installation is finished, don't forget to uncheck "Unknown sources".

How application works:

When started, application check is bluetooth enabled. If not, you will be warned and app. will enable bluetooth if you allow that. If you deny, app will stop and close itself. Also, app can check is BLE support available and warn you if not and, also, close itself if not. Application contains two main screens:

- <u>Scan screen:</u> here you will see all silverised devices in range (name, model and picture, depending on flashed firmware settings). Scanning for powered-on quads starts automatically when this screen is shown. Each scanning is performed for about 10 seconds and then app. stops with scan and you need to start it manually if you want to re-scan. When quads are listed, select the one you want to track telemetry by tap/click to open TLM screen.
- <u>TLM screen:</u> telemetry screen - contains data sent from selected quad

On both screens you can select "settings" (the usual way - by tapping/clicking on 3 dots placed in right corner in app bar) to set few options.

<u>Settings screen options:</u>

- Lowest batery voltage that will triger alarm (by default, it is set to 3.500V - the same value as default LVC value in firmware). Values you can set you choose between predefined ones that are common set of voltages inside secure limit for your battery.
- Lowest remote controller signal percentage that will cause alarm (default is 10% signal quality - also, you can turn this alarm off). .
- A way you want to show battery percetage - you can select percentage to be shown as real percentage depending on 3.270V as standard 1S LiPo "empty battery voltage" value (0% is when battery reach 3.270V or below - value 3.270 I found online on several pages as minimum voltage before battery is damaged and it is considered to be 0% of power - although LEDs work even below that as I tested with some bad battery I have...) or relative percentage depending on your low voltage alarm value (0% is when battery reach your low voltage value or below). Default is relative to your low value which means you will see percentage of battery voltage left before it drops to value set as low voltage for alarm).
- A way you want battery alarm to stop once it is triggered - by default, it is set to stop when voltage goes above defined limit - this means, for example, if low voltage alarm is set to 3.5V

and voltage drops below 3.5V (either because you really drain battery or maybe you aggressively hit throttle and you were near low voltage so it dropped to 3.5V), alarm starts and keep going while voltage is 3.5V or below. As soon as voltage goes up again (you reduced throttle, for example), it will stop. But if you want to be very careful to your batteries (😊) you can set alarm to stay always on once you hit defined limit - then once battery voltage drops to your defined limit or below, alarm will start and if battery goes back above the limit, alarm will still continue to alert you to stop flying and then only break in TLM signal (by either detaching battery from quad or going out of TLM screen or any other way you can turn TLM off) will then stop alarm no matter if voltage is recovered or not.

- Battery low voltage alarm type - a way you want to be alarmed when battery drops below defined limit - you can select between four different types of alarm: voice + red screen flashes (default), voice + flashes + vibration, flash + vibration, or red screen flashing only. Vibration pattern is repetitive long vibrations.
- Telemetry connection off alarm type - a way you want to be alarmed when TLM signal is lost (quadcopter out of bluetooth range or power-off etc.) - you can select between three different types of alarm: voice alarm + blue screen flashes (default), flashing + vibration (pattern is repetitive short vibrations) or flashing only.
- Telemetry detecting/receiving mode - you can set between three ways of BLE scanning/detecting/data receiving: mode 1 = auto-scan (default) - Android will scan for TLM data and send it to the app if BLE hardware in your Android device works this way. If not, you can choose mode 2 or 3 - both modes will start auto scan and occasionally (approx. every second) still try to force receiving TLM data - mode 2 will first stop auto scan and then restart it, while mode 3 will just try to start it again without stopping and BLE engine will prevent restart if it is already scanning (Huawei Honor 6 works best in mode 1, while XPeria T3 works OK in mode 2 and even better in mode 3).
- Initial screen on start - here you can select do you want PID data to be displayed instead part of telemetry data each time when you start tapplication, or not (details about PID data are further in text). Note that firmware you use must support transmitting PID values and PID tuning gestures support. Currently, all firmwares with SilverVISE support listed here have that feature.
- PID notation - select how you want PID values to be displayed. Select between standard decimal format and scientific notation.

TLM screen - what telemetry data can be tracked/seen and what alarms/messages you get:

- Model and visual interpretation of quad
- Power up time - how long your quad is turned on/connected to battery
- Flight time - how long you had motors spinning during one "session" - basically flying time until you disconnect battery (total time you was "in air" or had motors turned on - firmware code inspired by parts from bikemikem's FW version
- Motor status/"flying" info - only ON/OFF
- Information is quad bound to remote controller or not - useful if you change battery to remind you need to rebind - small info message that occurs once per bind, and small "chain" icon next to drone picture that will stay constantly on screen
- Visual info about currently set mode and rate/speed and audio indicator when you change mode/rate/speed (to easily hear changes you made when you flying - no need to look to app screen)
- Percentage, numeric and graphical display of battery status - current voltage, your defined low voltage that will cause alarm, % of battery voltage according to settings you set in "settings" screen
- Remote controller signal level with number of received packets per second (p/s) to have a feeling how good/bad is communication between remote controller and quad (are your quad too

far from you or maybe batteries inside controller are weak). Presented in raw numbers and percentage value,

- Remote controller RSSI data - actually totally "fabricated" and not calculated - based on some standard values for signal strength (in this case for current p/s value because values needed for RSSI calculation are not yet available in firmware). Not useable for anything except pure, "raw" (experimental) info.
- Currently selected type of battery voltage and TLM alarms
- Current status of bluetooth TLM connection
- Bluetooth RSSI data - value received from BLE hardware from your Android device and represent TLM signal strength between quadcopter and Android device (accuracy depends on BLE chip in your Android device)
- PID data - you can display or hide PID data by taping anywhere on screen. When displayed, PID data will hide some telemetry data (power-up time, flight time, motor status, RC signal level and RSSI) but all telemetry and warnings will still work. For more about PID data and setting PIDs using gestures in silverware firmwares, please read article here: http://sirdomsen.diskstation.me/dokuwiki/doku.php?id=pidgesture

Alarms:

- "Telemetry off" alarm (TLM alarm) - triggers if app lost connection to quad (if quad goes out of bluetooth range or it is turned off) for longer that 2 to 3 seconds (how you will be alarmed depends on what you set in settings - check settings for details) with automatically reconnect and continuing of checking when back in range or turn off/on quad. If connection is lost for a shorter period than 2-3 seconds, you will get info in status bar but alarm will trigger only if connection is lost for longer than 2-3 sec - so occasional brief dropouts of signal won't cause unnecessary alarms. If you experience constant TLM alarms and lost connections, please read "Possible problems you might get using app on your Android device and additional steps to check and solve_ckgedit_QUOT_. When TLM connections is back, you will be signaled by voice and/or vibration pattern (1 short vibration following 1 longer vibration), depending on TLM alarm style you set in settings
- "Telemetry off - duplicate name found" alarm - triggers if more than one quad with the same name is found transmitting in range of your Android device. App will stop telemetry and you need either to turn off other quad/quads or change name to more unique one and re-flash before use app with these quads.
- Remote controller signal (RC signal) low or lost alarm - beeping alarm to alert user when connection between RC and drone is lost (long beep) or drop below defined percentage (10% is initial value or user defined if set - and produce 4 short beeps that sounds like beeps from stock TX - these alarms are represented with beeps and not voice in case it overlaps with voice alarms so this way you can clearly differ warning sounds
- "Battery low" alarm if voltage drops below defined value - type of alarm depends on what you set in settings (check settings for details). This alarm has the higher priority than any other alarm - that means: if battery voltage gets below lowest set value and alarm triggers and, for example, bluetooth connection goes off after that, TLM alarm won't trigger - you will constantly get battery low warnings. Audible RC signal alarms will still be triggered. Note about battery voltage - if you check on-screen data during flight, you will see that voltage goes down-up-down-up as you fly (especially if you change speed much and push throttle a lot) - that is normal for LiPo batteries and when you get near your low voltage value you might get occasional alarms (or you will have constant alarm once you hit limit - depending on settings you set) if you push throttle hard because value might drop below limit (quad will also blink with LEDs triggering LVC during that dropouts). Then it's time to think about landing because you are getting near your low voltage value.

Info messages:

- Information is quad bound to remote controller or not - small info message that occurs once per bind
- Various mesages ("auto-scan in progress") etc

Warning messages:

- Battery low message
- Telemetry off
- Duplicate name found
- RC signal low (if not turned off in "settings" screen)
- RC signal lost

Possible problems you might get using app on your Android device and additional steps to check and solve:

If application does not geting any signal from the quad, check does your Adroid device have BLE capabilites (some models don't). If your Android device dos not support BLE, application should warn you. If app. did not warn you about BLE and you still have no connection to the quads, please try to scan for quad few times in a row because on some Android devices, some BLE chips during first finding of some device will "catch" signal only after few attempts (on these devices this delay and repetitive scanning need to be done only once and when quad is found, it will get its signal faster after that). Also, try to turn on Location Services (Settings → Location → On) - on some phones (notably the Nexus 5 phones), this is necessary step for BLE scanning to work (maybe your phone is also among them). If that does not help, install and use nRFConnect (https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en) to validate that your device support BLE and to exclude possibility that application have some problems that prevent you from using BLE: check does nRFConnect gets any TLM data when config.h is set as described in "How to prepare firmware..." section. If there is no data in nRFConnect, set firmware to work with standard Eddystone beacon by enabling RX_BAYANG_BLE like this:

```
//rx protocol selection
//#define RX_BAYANG_TELEMETRY
#define RX_BAYANG_BLE
//#define RX_BAYANG_BLE_APP
```

or (depending on firmware you use):

```
//#define RX_BAYANG_PROTOCOL_TELEMETRY
#define RX_BAYANG_PROTOCOL_BLE_BEACON
//#define RX_BAYANG_BLE_APP
```

Build and flash and then again check is there any packets available in nRFConnect.

If problem persists and your Android device still does not recognize your quad, you can reduce transmitting power in firmware by uncommenting appropriate lines (TX_POWER_GENERAL) in `rx_bayang_ble_app.c`. Also, you can force quadcopter to reduce power used for other work while TLM packet is in transmitting. Check `rx_bayang_ble_app.c` for all details (all entries are explained in details in `rx_bayang_ble_app.c`, but feel free to ask if you have any questions). Here is brief list of options that can help getting stable TLM connection:

- **TX_POWER_GENERAL**: if you have problem with TLM DISCONNECTED, try with default value or put here values from 0 - 7 - experiment while you get stable TLM connection without compromising controlling and flying quad
- **TX_POWER_ON_TLM**: reduces power during sending TLM data from the quad - also for better reception if you have problems receiving data from some boards to some Android devices and nothing else helps. Try with default value or put here values from 0 - 7 - experiment while you get stable TLM connection without compromising controlling and flying quad.
- **USE_ALL_BLE_CHANNELS**: TLM data will be transmitted on all 3 BLE channels (for better reception on some phones)

Now, happy flying and TLM-ing (or whatewer that might be called 🙂).
I hope this is application is usefull at least to some extent...