App-ShotManager communication

This document details App-Sololink communication for use in the Sololink behaviors.

Setup

ShotManager is the process that runs on Sololink that handles behaviors.
App-ShotManager communication happens over TCP on port 5507.

Protocol

All communication follows this TLV format in little endian:
Packet
{
	messageType : UInt32
	messageLength: UInt32
	messageValue : <n bytes as defined by messageLength>
}

We only have a few message types at the moment:

SOLO_MESSAGE_GET_CURRENT_SHOT - Sent from Solo to app when it enters a shot

messageType	0
messageLength	4
messageValue	Index of shot (SELFIE = 0, ORBIT=1, CABLECAM=2, FOLLOW=5)

SOLO_MESSAGE_SET_CURRENT_SHOT - Sent from app to Solo to request that Solo begin a shot

messageType	1
messageLength	4
messageValue	Index of shot (SELFIE = 0, ORBIT=1, CABLECAM=2, FOLLOW=5)

SOLO_MESSAGE_LOCATION - Sent from app to Solo to communicate a location

messageType	2
messageLength	20
latitude	Double
longitude	Double
altitude	Float (in meters)

SOLO_MESSAGE_RECORD_POSITION - Sent from app to Solo to request recording of a position

messageType	3
messageLength	0

SOLO_CABLE_CAM_OPTIONS - Sent from app to Solo or vice versa to transmit cable cam options

messageType	4
messageLength	8
camInterpolation	Short - 0 if interpolation is off 1 if on
yawDirection	Short - 0 means counter clock wise 1 means clockwise
cruiseSpeed	Float (in meters/second)

SOLO_GET_BUTTON_SETTING - Sent from app to Solo to request Button mapping setting. Sent from Solo to app as a response

messageType	5
messageLength	16

button	Int32 - ButtonPower = 0 ButtonFly = 1 ButtonRTL = 2 ButtonLoiter = 3 ButtonA = 4 ButtonB = 5 ButtonPreset1 = 6 ButtonPreset2 = 7 ButtonCameraClick = 8
event	Int32 - Press = 0 Release = 1 ClickRelease = 2 Hold = 3 LongHold = 4 DoubleClick = 5
shot	Int32 - shot index, -1 if none. One of shot/mode should be -1, and the other should have a value
mode	Int32 - APM mode index, -1 if none

SOLO_SET_BUTTON_SETTING - Sent from app to Solo to set a Button mapping setting.

messageType	6
messageLength	16
button	Int32 - ButtonPower = 0 ButtonFly = 1 ButtonRTL = 2 ButtonLoiter = 3 ButtonA = 4 ButtonB = 5 ButtonPreset1 = 6 ButtonPreset2 = 7 ButtonCameraClick = 8
event	Int32 - Press = 0 Release = 1

	ClickRelease = 2 Hold = 3 LongHold = 4 DoubleClick = 5
shot	Int32 - shot index, -1 if none. One of shot/mode should be -1, and the other should have a value
mode	Int32 - APM mode index, -1 if none

SOLO_SHOT_OPTIONS - Sent from app to Solo or vice versa to transmit selfie options

messageType	20
messageLength	4
cruiseSpeed	Float (in meters/second)

These messages only go from Shot Manager -> App

SOLO_SHOT_ERROR - Sent from Solo to app when entry into a shot was attempted but rejected due to poor EKF

messageType	21
messageLength	4
errorType	BAD_EKF = 0, UNARMED = 1

SOLO_MESSAGE_SHOTMANAGER_ERROR - Debugging tool - shotmanager sends this to the app when it has hit an exception

messageType	1000
messageLength	N (length of exceptStr)
exceptStr	Exception info and stacktrace

SOLO_CABLE_CAM_WAYPOINT - Send the app our cable cam waypoint when it's recorded

messageType	1001
messageLength	28
latitude	Double
longitude	Double
altitude	Float (in meters)
degreesYaw	Float (yaw in degrees)
pitch	Float (camera pitch in degrees)

Flows

Cable cam

Either the app can request that cable cam starts (via SOLO_MESSAGE_SET_CURRENT_SHOT) or the user can initiate from Artoo, in which case shotManager will tell the app via a SOLO_MESSAGE_GET_CURRENT_SHOT. This only works if an app is connected to shot manager.

To proceed, the user needs to record two locations.

The app can tell shotManager to record a point using SOLO_MESSAGE_RECORD_POSITION, but it should always wait to receive a SOLO_CABLE_CAM_WAYPOINT before proceeding.
Otherwise, a user can Press 'A' to record a point on Artoo.

If a user tries to record two points on top of each other, the second one overwrites the first. To inform the app, ShotManager will send a SOLO_MESSAGE_GET_CURRENT_SHOT with -1 as a shot which should tell the app to exit cable cam, then a SOLO_MESSAGE_GET_CURRENT_SHOT with 2 to reenter cable cam, and then a SOLO_CABLE_CAM_WAYPOINT to reflect the new starting point of the cable.

At any point, if the user hits ‘B’, it will exit cable cam and shotManager will send a SOLO_MESSAGE_GET_CURRENT_SHOT with -1 to the app.

After the second point is recorded, shotManager will send a SOLO_CABLE_CAM_OPTIONS to the app so the app can retrieve the memorized yaw direction. The app can send SOLO_CABLE_CAM_OPTIONS to shotManager to change any of the options. If the user hits the “Pause” button on Artoo, shotManager will adjust cruiseSpeed to 0.0 and send SOLO_CABLE_CAM_OPTIONS to the app. Hitting “pause” again will set cruiseSpeed to the original speed and send it to the app.

When the second point is recorded, shotManager will set Solo to Guided and the copter will be on the cable.

Selfie

Selfie is much simpler than cable cam.

The flow must be initiated by the app via a SOLO_MESSAGE_SET_CURRENT_SHOT. ShotManager then expects 3 locations sent using SOLO_MESSAGE_LOCATION. They are:

- 1st selfie waypoint (near point)
- 2nd selfie waypoint (far point)
- selfie ROI point

Upon receiving these 3 points, shotManager will put Solo into guided mode and the selfie will automatically start. It’s controllable and pausable just like cable cam though.

User can press ‘B’ to exit as in cable cam.

Button mapping:

The button mappings for ‘A’ and ‘B’ are stored on Solo. The app should poll these settings using SOLO_GET_BUTTON_SETTING, upon which shotManager will send the current setting back.

If a user changes the setting, the app can set it with SOLO_SET_BUTTON_SETTING.

We only use “Press” events at the moment.

Orbit

Either the app can request that orbit starts (via SOLO_MESSAGE_SET_CURRENT_SHOT) or the user can initiate from Artoo, in which case shotManager will tell the app via a SOLO_MESSAGE_GET_CURRENT_SHOT. This only works if an app is connected to shot manager.

To proceed, the user needs to lock onto a spot. This can be done in 3 ways:

- Press app banner. This sends a SOLO_RECORD_LOCATION to shot manager, where Orbit will record its current ROI.
- Press 'A' on Artoo. Orbit will record its current ROI.
- Long press on the map on the app. This will send a SOLO_MESSAGE_LOCATION to shot manager, which will be the ROI.

In any case, if an initial ROI is set for Orbit, it will send this location back to the app via a SOLO_MESSAGE_LOCATION message.

At any point, if the user hits 'B', it will exit orbit and shotManager will send a SOLO_MESSAGE_GET_CURRENT_SHOT with -1 to the app.

The app can send SOLO_SHOT_OPTIONS to shotManager to change cruise speed. If the user hits the "Pause" button on Artoo, shotManager will adjust cruiseSpeed to 0.0 and send SOLO_SHOT_OPTIONS to the app. Hitting "pause" again will set cruiseSpeed to the original speed and send it to the app.

When an initial ROI is set, shotManager will set Solo to Guided and the copter will be in orbit mode.

During orbit mode, the app can send a SOLO_MESSAGE_LOCATION to update the ROI location and begin a new orbit around the new ROI. Note this differs from Follow mode, which sends new points but does not reset the orbit.

Prior to the ROI being set, the app should show on the app the projected ROI of orbit. This is not passed from shot manager to the app, so the app should calculate it. This calculation works as follows:

```
if camera.getPitch(self.vehicle) > SHALLOW_ANGLE_THRESHOLD (-60):
        loc = location_helpers.newLocationFromAzimuthAndDistance(self.vehicle.location,
camera.getYaw(self.vehicle), FAILED_ROI_DISTANCE  (20.0))
    else:
        loc = roi.CalcCurrentROI(self.vehicle)
```

Buttons for cruising work as they do in cable cam.

Follow:

Follow is a special case of orbit. It works the same way except instead of setting or locking onto an roi, it uses the phone's GPS. In order to do this, it needs to connect to a udp port on shot manager.

First the app should tell shot manager to enter the Follow shot via SOLO_MESSAGE_SET_CURRENT_SHOT. Follow is not enterable via Artoo.

Then the app should begin to stream positions in SOLO_MESSAGE_LOCATION packets to shot manager on port 14558.

Buttons for cruising work as they do in cable cam.