TALARC AlP40K – Design and Construction

Goal

To design a Pico-Balloon that could fly at or above 40,000' with a payload of 40 – 50 grams, operate 24 hours a day and do Science.

Design

After doing research on the groups.io pico-ballooning site and other mixed media, it became evident that to be successful, it would require a battery that could handle the cold temperatures encountered at high altitudes. After a lot of searching, I found and selected a Tadiran Lithium battery **TLI-1550ES**. Though not perfect, it provides the best operating temperature range, capacity and mass of any I found. Additionally, I would use a similar platform to what was used for **ALP40-G** that circled the globe. Note: A new Ethyl Acetate chemistry looks promising for the future.

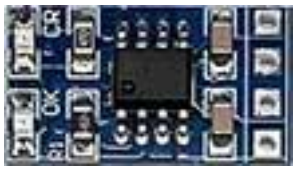
ALP40-K

After the two-week success of ALP40-G, I used the same design for ALP40-K, but with a sealed battery compartment; a battery compartment temperature sensor, Innovative Sensor Technology **P10K.520.6W.B.010.D**; IR temperature sensor, **TBP-I2C-H04** (4° field of view); a UV Sensor **GUVA S-12SD**; a **3.7 4.2 MPPT Solar controller** for a Lithium battery charging and a higher power solar panel, Anysolar IXOLAR High Efficiency Solar **MD SM141K08TF**. I flew QRP Lab latest version of the **U4B/firmware**, which added extended telemetry (35 bits) and modified I2C command word. The U4B BASIC flight software was modified to transmit alternate IR sensor temperature and battery compartment temperature. A slot 2 WSPR extended telemetry transmission provided IR sensor temperature, battery compartment temperature, altitude least significant two digits (altitude to 1 meter accuracy), UV Index sensor, solar panel voltage and CPU reset flag (24 hour). I used a standard GPS and HF (20 m) dipole antennas.

Hardware Photos Below



UV Light Sensor



MPPT Solar Charger



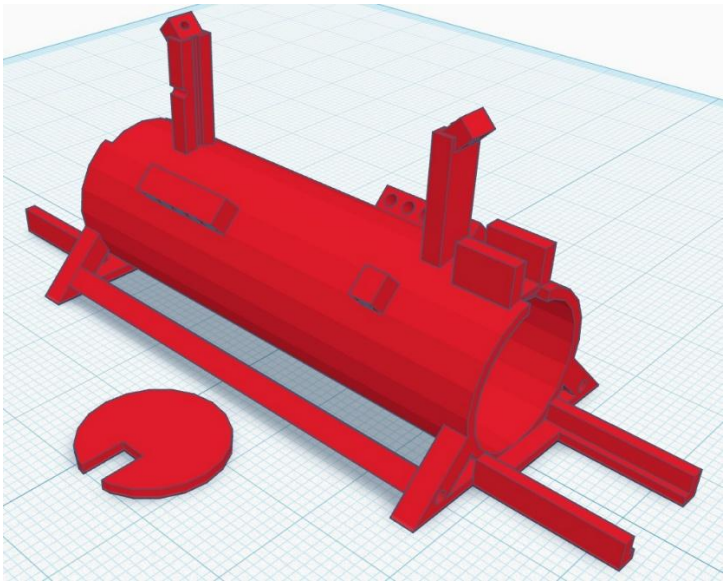
IR temperature sensor, TBP-I2C-H04



TLI-1550A/Z2/T



Solar Panels and Schottky Diodes



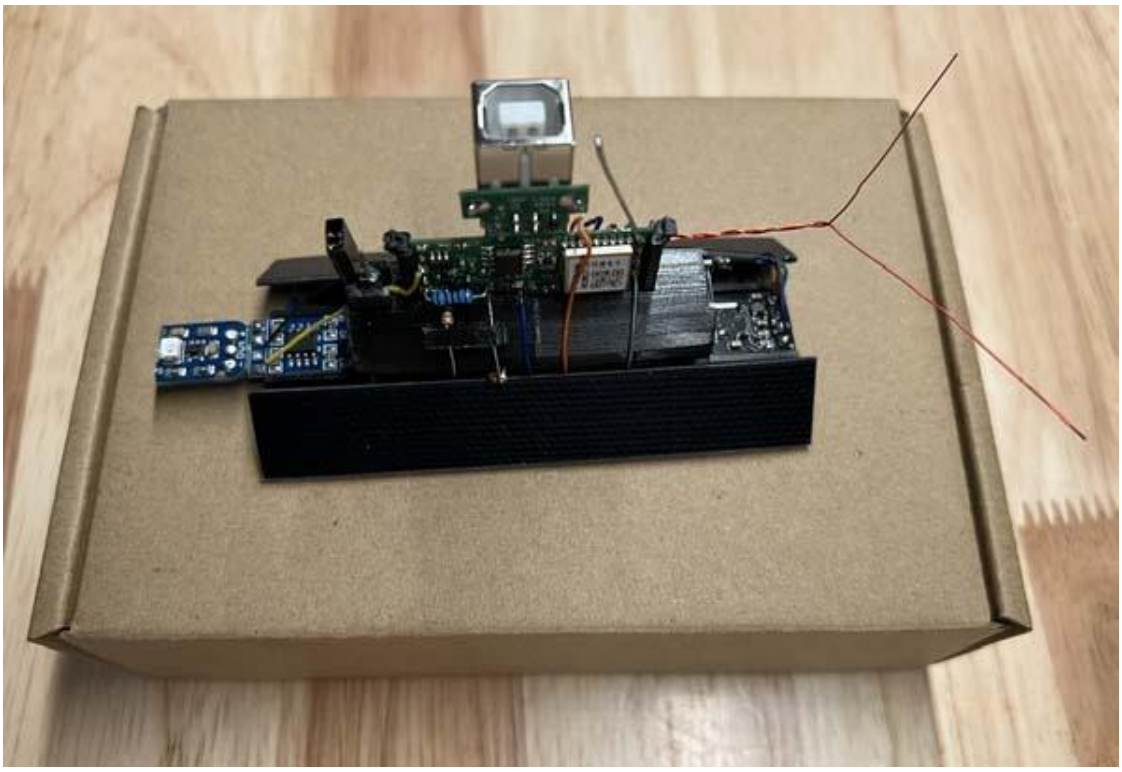
3D printed frame using ABS plastic

I used a **Cymylar 60** (Name not diameter) balloon that was stretched to 115”, shown below.





The Balloon Filled with Hydrogen, Heat Sealed and Taped with 20 m Dipole Antenna Attached



ALP40-K Payload Almost Ready to Fly (USB Connector Needs to Be Removed and Antenna Attached)

Experiments and Data to be Collected

U4B Standard Slot 0 and 1 Telemetry
Custom Telemetry Slot 2 (35 bits)
Battery Compartment Temperature
IR Temperature Sensor Pointed Toward Earth
UV Sensor
MPPT Solar Charging Circuit
Lithium Battery Viability
20m Dipole Antenna VSWR Tested at 1.5:1
CW Transmissions Reverse Beacon Net

TALARC ALP40-K Information and Telemetry

WSPR Slot 0 (2-minute transmission)
Date/Time
Callsign Transmitted
Grid Square 4
Reception Frequency
Received SNR
Frequency Drift
Transmitter Power
Reporter Callsign/Grid Square 6
Range/Azimuth

WSPR Slot 1 (2-minute transmission)

Date/Time
Grid Square 4 + 2
Altitude
Speed
PCB Supply Voltage
PCB Temperature
Sun Angle (Calculated from Grid Square 6)

Decode of Extended Telemetry

TALARC ALP40-K Generic Extended Telemetry is decoded from “Big Number” transmitted **WSPR Slot 2 (2-minute transmission)** as follows:

As an Example, Let Big Number = 12592717190

Solar Panel Voltage (6)	Battery Temperature (7)	IR Object Temperature (7)
0 1 0 1 1 1	0 1 1 1 0 1 0	0 1 0 1 0 1 1
23	58	43

UV Index x 4 (7)	Altitude Least Sig 0-99 (7)	CPU Reset Flag (1)
1 1 0 0 1 1 1	1 0 0 0 0 1 1	0
103	16	0

Solar Panel Voltage (Volts) = MOD(INT(BigNumber/2/128/128/128/128),64)=23 → Lookup in Table=3.95 Volts
Battery Temperature (Kelvin)* = MOD(INT(BigNumber/2/128/128/128),128) + 200 = 258K
IR Object Temperature (Kelvin)* = MOD(INT(BigNumber/2/128/128),128) + 200 = 243K
UV Index = MOD(INT(BigNumber/2/128),128) / 4 = 25.75 UV Index
Altitude Least Significant 0 – 99 +/-1.5 meters (M) = MOD(INT(BigNumber/2),128) =16 m
U4B Reset Flag (T/F)** = MOD(BigNumber,2) = Has not reset

*Note: Temperature range 200K (-73°C or -99°F) to 327K (54°C or 129°F)
**Note: If TRUE, CPU has been reset in the past 24 hours

Value	Volts	Value	Volts
0	2.80	32	4.40
1	2.85	33	4.45
2	2.90	34	4.50
3	2.95	35	4.55
4	3.00	36	4.60
5	3.05	37	4.65
6	3.10	38	4.70
7	3.15	39	4.75
8	3.20	40	4.80
9	3.25	41	4.85
10	3.30	42	4.90
11	3.35	43	4.95
12	3.40	44	5.00
13	3.45	45	5.05

14	3.50	46	5.10
15	3.55	47	5.15
16	3.60	48	5.20
17	3.65	49	5.25
18	3.70	50	5.30
19	3.75	51	5.35
20	3.80	52	5.40
21	3.85	53	5.45
22	3.90	54	5.50
23	3.95	55	5.55
24	4.00	56	5.60
25	4.05	57	5.65
26	4.10	58	5.70
27	4.15	59	5.75
28	4.20	60	5.80
29	4.25	61	5.85
30	4.30	62	5.90
31	4.35	63	5.95

Software

Standard QRP Labs U4B BASIC programming was used. Program follows use at your own risk as it may contain errors.

TALARC Balloon AP40-J Flight Code

Configuration

KB7HTA-18 – 20 Meters – CH 250 – WSPR 14.097140 – CW 14.022000 – Start Minute 08/10 – CS 1?2

Variables

Letter	Initial	Description
S	1	Battery Set Point S = 6 (> 3.9 V), S = 3 (3.7 – 3.9 V), S = 2 (3.4 – 3.7 V) S = 1 (< 3.4 V)
H	3800	This is the battery “High” set point in millivolts.
L	3600	This is the battery “Low” set point in millivolts.
V	3400	This is the battery “Very Low” set point in millivolts.
U	0	UV Sensor index
O	273	This holds the battery compartment temperature in Kelvin after the “BCT” subroutine is RUN
F	0	Flag for alternating Battery and PCB temperature, and TELEX Slot 3 and CW CQ.
G	0	GPS Start second(s) Reset
J	0	GPS End second(s) Running
E	0	Next Day Flag
I	0	IR Sensor temperature channel I = 6 for ambient, I = 7 for object
M	2500	Below this altitude in meters, send a CW report with latitude, longitude and battery voltage
N	0	Most recent GPS lock time
Y	0	This variable holds the result of reading the solar panel voltage after the diodes but before the Lithium battery charging circuit. There is a 10K Ohm divider network to cut

		the voltage in half to keep the voltage within the voltage range of the ADC. The full scale 12- bit ADC reading would be 4096 representing 3.3 Volts. The software multiplies this number by 2 to restore the full range of voltages 0 – 6.6, lost due to the voltage divider.
P	0	This holds the 35.8 Bit Big Number for extended telemetry slot 2
R	0	CPU Reset in Last 24 Hours
Z	0	IR Sensor Temperatures
A	?	Temporary Variable
B	?	Temporary Variable
C	?	Temporary Variable
D	?	Temporary Variable

```
LET FR = 25000000
LET CH = 250
LET HP = 1
LET S = 1
LET N = 0
LET P = 0
LET R = 0
LET E = 0
LET Z = 0
LET H = 3800
LET L = 3600
LET V = 3400
LET O = 273
LET F = 0
LET G = 1
LET J = 0
LET U = 0
LET M = 2500
GPS 300                          / Check GPS to make sure it is working
RUN “QLOCK”                              / Flash LED 10 time and leave LED on if no lock --- Abort launch
LET G = VAL "#2.1TM"             / Let G be the initial time in hours
LET G = G * 60 + VAL "#3.2TM"    / G converted to minutes
LET G = G * 60 + VAL "#5.2TM"     / G converted to seconds
10 GPS 200
RUN “BSPS”                       / Set battery set point
IF R = 1
RUN “ADAY”
ENDIF
OUT 19 0
SLEEP 10 7
OUT 19 1                         / Use OUT 19 0 for flight to conserve power
RUN “QBCT”
RUN “PAC2”
IF F = 0
CW 0 14022000 12 0 “CQ KB7HTA KB7HTA BALLOON #M6 #AT #BT”
ENDIF
```

```
TELE
TELEC P
IF AT < M        ←Sends Latitude Longitude and battery voltage if below the value of M (2500 meters) on 2m
CW 0 146585000 12 0 “KB7HTA #LT #LN #LT #LN #BT”
ENDIF
GOTO 10
END
~
```

BSPS – Determine the Battery Set Point S = 6 (> 3.8 V), S = 3 (3.6 – 3.8 V), S = 2 (3.4 – 3.6 V) S = 1 (< 3.4 V). The variable S specifies the number of times a WSPR transmission is made each hour. Use the Battery Set Point to determine how often to send WSPR transmissions by introducing CPU sleep cycles.

```
LET S = 1
IF BT > V
   LET S = 2
ENDIF
IF BT > L
   LET S = 3
ENDIF
IF BT > H
   LET S = 6
ENDIF
IF S = 1
   LET A = 5
   RUN “SLPX”
ENDIF
IF S = 2
   LET A = 2
   RUN “SLPX”
ENDIF
IF S = 3
   LET A = 1
   RUN “SLPX”
ENDIF
~
```

SLPX – Sleep for the number of transmission cycle(s) determined by the variable A.

```
OUT 19 0
FOR X = 1 TO A
   SLEEP 10 7
NEXT
OUT 19 1
~
```

QBCT – Determine If **B**attery **C**ompartment **T**emperature or IR Sensor Temperature Needs to be Read and Substituted for U4B Temperature. A value of 2 K was added to 'O' and 'Z' the value stored in the system variable TT to take care of round error between the encoding done by PAC2 and the encoding done on system variable.

```
IF F > 0
   RUN "BCT"
   LET TT = O + 2
   LET F = 0
ELSE
   LET I = 7
   RUN "RIRT"
   LET TT = Z + 2
   LET F = 1
ENDIF
~
```

BCT – Read and Convert the **B**attery **C**ompartment **T**emperature to Kelvin. The resulting temperatures seem to be 7 K higher than they should be, so 7 is subtracted from the value before leaving the subroutine

```
LET A = INA 4
DELAY 1
LET A = INA 4
LET B = A * 806
LET C = 3300000 – B
LET D = C / ( B / 10000 )
LET O = ( D / 38 ) + ( D / 39 )
LET O = ( O / 2 ) - 7
~
```

FLASH – This program is used to send the number of light flashes specified by the variable A.

```
FOR X = 1 TO A
OUT 19 0
DELAY 1000
OUT 19 1
DELAY 1000
NEXT
~
```

QLOCK – This program determines if the balloon has a GPS lock. If there is a lock it turns off the LED, otherwise it flashes the LED 10 times and leaves it lit.

```
IF GL = 0
LET A = 10
RUN "FLASH"
OUT 19 1
ELSE
OUT 19 0
```

```
ENDIF
~
```

RIRT – Read the IR Temperature Sensor

```
LET Z = 0
FOR X = 1 TO 5
   LET D = I2CR 117 7 3
   LET D = D % 65536
   LET Z = Z + D
NEXT
LET Z = 2 * Z / 500
~
```

RSPV – Read the solar panel voltage after the diodes.

```
FOR X = 1 TO 5
LET Y = Y + INA 7
DELAY 10
NEXT
LET A = 0
LET Y = Y / 5
LET Y = Y * 1000 / 4096
LET Y = Y * 3300 / 10000
LET Y = Y * 2
IF Y > 280
   LET Y = (Y – 280) / 5
ELSE
   LET Y = 0
ENDIF
~
```

PAC2 – Package extended telemetry SLOT 2, solar panel voltage, battery compartment temperature (K), IR object temperature (K), IR ambient temperature (K) and last two LS Digits of the balloon's altitude (m) and CPU reset last 24 hours.

```
LET P = 0
RUN "RSPV"
LET P = Y * 128
RUN "BCT"
LET P = ( P + ( O – 200 ) ) * 128
RUN "RIRT"
LET P = ( P + ( Z – 200 ) ) * 128
LET A = INA 2                    /Read UV sensor raw value 0 to 4096 decimal
DELAY 1
LET A = INA 2                    /Read UV sensor raw value 0 to 4096 decimal
LET U = A / 32                   /Convert to fit into 7 bits 0 to 128 decimal
LET P = ( P + U ) * 128
```

```
LET A = AT + 100000                 / add 100k to altitude make sure string has 6 digits for next step
LET B = VAL “#5.2VA”                / grab the last 2 digits of altitude
LET P =  ( P +  B )  * 2
LET P = P + R
~
```

ADAY – Has a day passed since a system reset

```
LET A = 0
LET B = 0
LET J = VAL "#2.1TM"                / Let J be the end time in hours
LET J = J * 60 + VAL "#3.2TM"       / J converted to minutes
LET J = J * 60 + VAL "#5.2TM"       / J converted to seconds
IF J < G
LET E = 1
ELSE
LET A = 1
ENDIF
LET B = A + E
IF B = 2
LET R = 0
ENDIF
~
```

Balloon Mass Analysis

3D ABS Print w/SP Mount:	4.00 grams
Solar Battery Charger:	0.70grams
IR Temperature Sens:	2.38 grams
Battery Temp Sensor:	0.01
U4B Tracker:	1.80 grams
UV Sensor:	1.00
Solar Panels x 2:	6.16 grams
Battery:	21.85 grams
Miscellaneous:	2.60 grams
Payload Mass:	40.50 grams
Antenna Top:	1.60 grams- #34-gauge copper wire 16’ 7.2”, 6# fishing line & silicone
Antenna Bot:	1.00 grams- #34-gauge copper wire 16’ 7.2”
Total Payload:	43.10 grams
Float:	5.00 grams
Neck Weight:	48.10 grams