

Publicámos nas revistas QSP de Agosto de 2015 e na de Outubro de 2018 algumas notas sobre o lançamento de Balões estratosféricos não esquecendo os falhanços e as peripécias

# Pay Load para o Balão sonda (ATECA-2) com o Raspberry PI



Por: **Carlos Miranda**  
CT4BB

Nesses artigos referimos quais os procedimentos legais a realizar para se lançar um balão sonda, porque é uma matéria legislada que carece de licença. Embora o assunto aqui exposto possa ser um tanto complicado e maçador, penso que alguns colegas “mais entusiastas destas coisas” possam retirar deste artigo e dos programas baixados da Internet algumas ideias úteis para os seus projectos.

Tal como prometemos, aqui estamos para explicar a constituição do hardware e filosofia do software do Pay Load, porque o coração do sistema é um Raspberry PI.

## OBJECTIVO DO PROJECTO

Depois de uma interrupção de dois anos

***Foi adquirida e instalada uma placa GPS ADAFRUIT que funciona até 40 Km de altitude. A maioria das placas GPS do mercado dificilmente ultrapassam os 10 Km por razões de segurança internacional contra o desenvolvimento privado de projecteis bélicos.***

por falta de verba, tempo e COVID, para nos dedicarmos a 100% ao desenvolvimento do hardware e do software, voltamos agora para mais uma tentativa.

O Grupo de Radioamadores GRC de Cascais, já lançou 3 Balões experimentais ATECA 0A, 0B e 1 integrados no projecto ATECA - Aeróstato Transportando Experiencias de Comunicações de Amador destinado ao estudo e comportamento dos equipamentos electrónicos de medição e de

comunicações durante a travessia das camadas da atmosfera e da estratosfera sendo que, o maior interesse, para nós, é o de utilizar um Raspberry Pi 3 a comandar os interfaces, realizando todas as funções de medição, emissão e gestão de dados: GPS, Medição de temperaturas, Conversão ADC,

Fotografia, geração de RF, Detecção de raios gama, e actuar uma sinalização.

O Raspberry Pi 3 foi o escolhido, porque pesa apenas 65 gramas, tem uma grande capacidade de processamento e, ainda tem a vantagem de existirem programas na Internet sobre as suas aplicações.

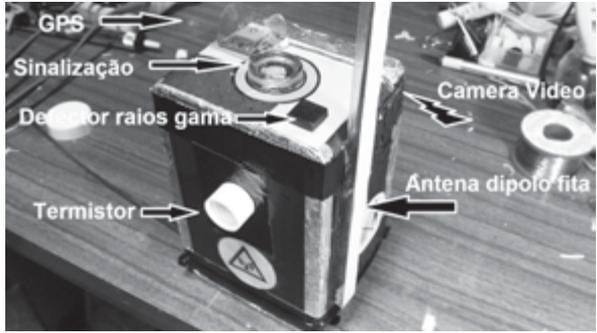
Só há que associá-los e pô-los a fazer o que queremos

**O Raspberry Pi executará as seguintes tarefas:**

**1.1 – Captar imagens** da atmosfera/estratosfera através de uma Camera exterior, enviando-as logo de seguida para terra em SSTV na frequência de 144.430 MHz.

**1.2 - Armazenar as fotografias em HD** no cartão SD do Raspberry Pi.

**1.3 -Transmitir em APRS, nos 144.800 MHz,** para uma iGate em terra, as temperaturas interior e exterior, a pressão atmosférica, a altitude, o número de impulsos de raios gama,



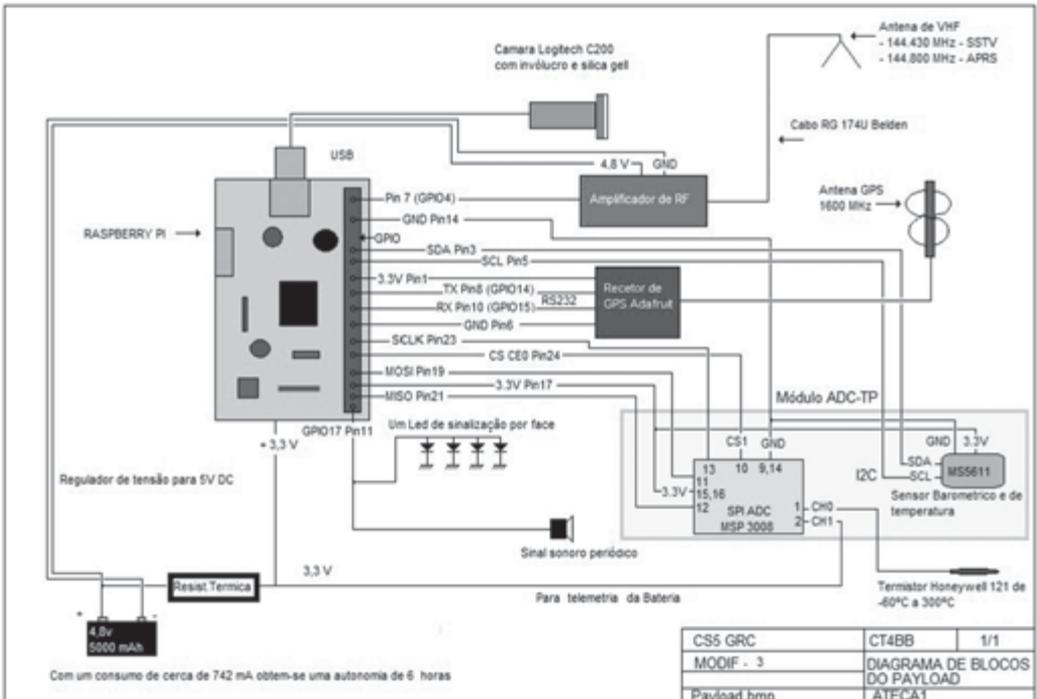
**Figura 1** Aspecto exterior do Pay Load

o estado das baterias e as coordenadas/altitude. A iGate em terra, envia os dados para o Google/Aprs.fi que os regista e elabora os respectivos gráficos.

**1.4 – Transmitir em fonia e em tempo real** na frequência de 144.550 MHz, a altitude em que se encontra o balão e o sentido do movimento - a subir ou a descer.

**1.5 - Fazer a sinalização** através de um

**Figura 2** Diagrama de Blocos do Hardware do Pay Load



alarme audiovisual para facilitar a localização e recolha após a queda.

*Para interface com Raspberry, realizámos uma placa de Conversão Analógica Digital mostrada na Figura 4 que faz a conversão para digital das medições analógicas.*

Foi adquirida e instalada uma placa GPS ADAFRUIT que funciona até 40 Km de altitude. A maioria das placas GPS do mercado dificilmente ultrapassam os 10 Km por razões de segurança internacional contra o desenvolvimento privado de projecteis bélicos.

Foi construído um amplificador de RF para VHF com 200 mW de saída equipado com um GALI-84 da Mini-Circuits que amplifica a RF gerada pelo Raspberry Pi nas frequências de 144.550 MHz para a fonia, 144.800 MHz para o APRS e em 144.430 para SSTV.

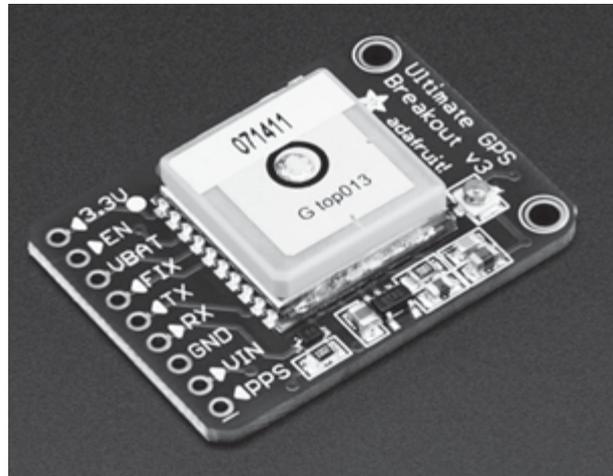
A emissão SSTV do Pay Load devia ser programada para transmitir nos 144.500 MHz, que é a frequência Standard de chamada em SSTV. Mas, como a emissão do Pay Load é intempestiva e a frequência pode estar ocupada, decidimos fazer as emissões de SSTV nos 144.430 MHz para não interferir com as emissões de SSTV em terra.

## **2 - FUNÇÕES DO RASPBERRY PI E INTERFACES**

### **2.1 – Captação e emissão das coordenadas**

As coordenadas e a altitude obtidas da placa GPS Figura 3, são enviadas para um buffer de onde são transmitidas em APRS para terra. Esta informação, via iGate, fica registada no Aprs.fi.

Os dados registados servirão para o traçado do percurso num mapa do Google Aprs.fi em tempo real e determinar com aproximação o local da queda.



*Figura 3 - Placa GPS (Adafruit) para 40.000 metros*

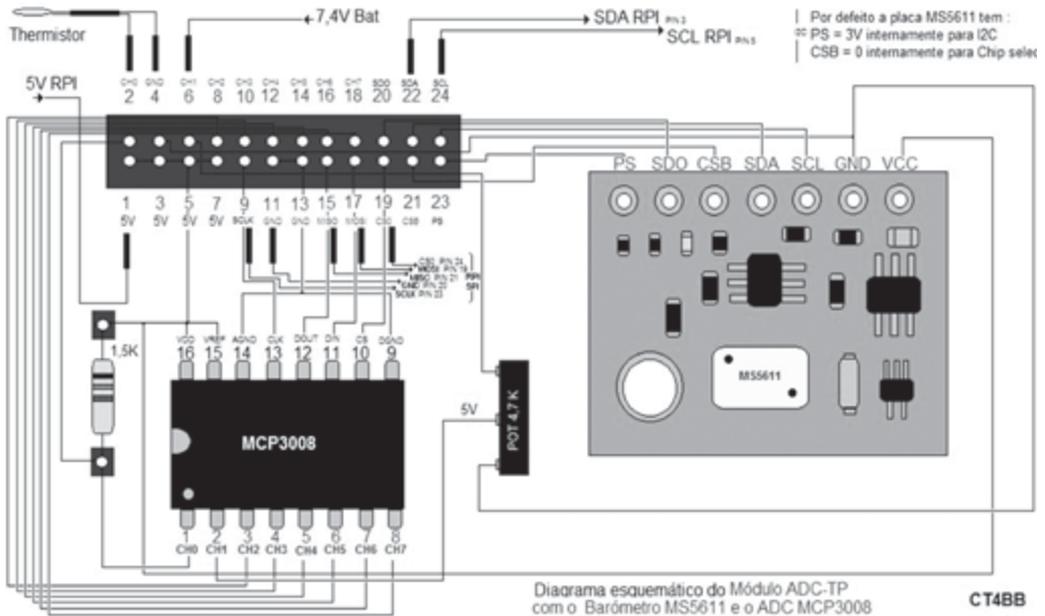
*Nos três lançamentos anteriores verificámos que as coordenadas do local da queda tinham um erro relativamente grande, porque o sinal de RF do Pay Load à iGate não chegava, quando a altitude da iGate é inferior a uns 300 metros. Esta é a razão pela qual colocamos a iGate num ponto alto.*

Por outro lado, mesmo com a iGate a boa altitude, a emissão do Pay Load com apenas 200mW no meio de um campo de milho é muito difícil de receber, porque acontece normalmente a 100Km da iGate. Mesmo a radiolocalização é difícil. Para resolver este problema vamos utilizar um SPOTTRACE que faz o rastreio via satélite com uma “app” no telemóvel.

### **2.2 – Captação de imagens**

Para a captação de imagens utilizamos a camera a cores acessória do Raspberry Pi.

O software de configuração e activação da



**Figura 4 - Módulo de conversão ADC SPI e I2C**

camera foi baixado da Internet a partir do servidor de aplicativos do Raspberry Pi via linha de comandos.

Não optamos pelas Camaras PC/WEB, porque são muito sensíveis à luz.

São desenhadas para funcionar em interiores com iluminação reduzida. Mesmo utilizando filtros UV para resolver a saturação não obtivemos resultados satisfatórios. Por outro lado, o peso de uma camera WEB é 100 vezes superior ao da Camera do RASPI.

### 2.3 – Medida das temperaturas interior/ exterior e pressão atmosférica

A medida da temperatura interior e exterior ao Pay Load dá-nos uma visão do comportamento térmico interior e exterior do módulo durante o percurso, em função da altitude e das hora:minuto:segundo.

A temperatura exterior é obtida a partir de um termistor e a temperatura interior a partir de um sensor no interior do IC MS5611. Este integrado dá também o valor da pressão atmosférica.

*Podemos assim também vigiar o comportamento térmico interior do Pay Load, porque a maioria dos dispositivos electrónicos nomeadamente as baterias, não funcionarão com temperaturas abaixo de -10 °C.*

O valor analógico medido pelo termistor é convertido em digital através do IC conversor ADC MCP3008.

O termistor dá temperatura exterior num valor analógico que é convertido em digital pelo MCP3008 este, por sua vez, envia o valor digital para o Raspberry Pi por comunicação SPI. A temperatura interior e a pressão atmosférica que são obtidas do MS5611 saem no formato digital e são enviadas para o Raspberry Pi em comunicação I2C. Figura 4

A medida da pressão atmosférica é im-

portante, porque permitirá analisar o comportamento do Balão e determinar qual foi a pressão atmosférica no ponto de rebentamento. Será também interessante comparar a altitude calculada a partir de pressão atmosférica com a altitude medida pelo GPS.

### 2.5 – Medição da carga da bateria.

Medir o comportamento da carga da bateria durante o percurso vai permitir registar num gráfico no Aprs.fi a variação da tensão ao longo do percurso e saber se as falhas do sistema electrónico foram ou não causadas por essas variações. A alimentação de tensão dos circuitos electrónicos é feita através de um fio cromoníquel "Resistência Térmica" mostrada no esquema da Figura 2 que é enrolado em volta das baterias, aquecendo-as ligeiramente, para evitar que o arrefecimento as torne ineficazes.

## 3 - ESTRUTURA DAS MEDIÇÕES E O FORMATO DO APRS

*O Raspberry Pi e os interfaces farão 11 medidas:* umas obtidas do GPS e outras dos sensores.

Os dados serão transmitidos de 3 em 3 minutos intercalados com as imagens de SSTV.

*Serão transmitidas 3 Tramas seguidas de APRS:* duas com dados do GPS e uma com dados dos sensores.

*As tramas emitidas em APRS têm o seguinte formato:*

C5ATEC-11>APRS: \$GPRMC,151447,A,4034.5189,N,10424.4955,W,6.474,132.5,220,406,10.1,E\*58

C5ATEC-11>APRS: \$GPGGA,151449,4034.5163,N,10424.4937,W,1,06,1.41,21475.8,M,-21.8,M,,\*4D

T#000,245,123,45,24,252

*Destas tramas extraem-se:*

1- Hora UTC extraídas da trama (GPGGA)

2- Coordenadas geográficas extraídas da trama (GPGGA)

**Medir** o comportamento da carga da bateria durante o percurso vai permitir registar num gráfico no Aprs.fi a variação da tensão ao longo do percurso e saber se as falhas do sistema electrónico foram ou não causadas por essas variações.

3- Altitude GPS acima do nível do mar da trama (GPGGA)

4- Azimute do percurso no momento da trama (GPRMC)

5- Velocidade horizontal em metros em Km/Hora da trama (GPRMC)

6- Variação Magnética da trama (GPRMC)

*A telemetria será transmitida de 9 em 9 minutos em 3 tramas com a seguinte formatação APRS:*

:C5ATEC-11 :PARM.TempInt,TempExt,Press,AltiP,Velsub (Parâmetros)

:C5ATEC-11 :UNIT.Graus. C, Graus. C,HcPasc,Metros,m/seg (Unidades)

:C5ATEC-11 :EQNS.0,82,0,0,2.4,0,0,4.2,0,0,0,0,0,6,0 (Coeficientes)

*As Telemedidas serão de:*

7- Temperatura interior dos equipamentos (TempInt)

8- Temperatura exterior (TempExt)

9- Pressão atmosférica (Press)

10- Altitude calculada a partir da pressão atmosférica pelo SW(AltiP)- para se estudar e comparar com a altitude do GPS, cujo erro aumenta a partir do 10.000m.

11- Velocidade de subida calculada pelo SW (Velsub)

*Estes valores serão também transmitidos de 3 em 3 minutos entre cada imagem SSTV e no formato:* T#000,245,123,45,24,252.



## PLANO DE TRANSMISSÕES DA SONDA ESTRATOSFÉRICA ATECA1

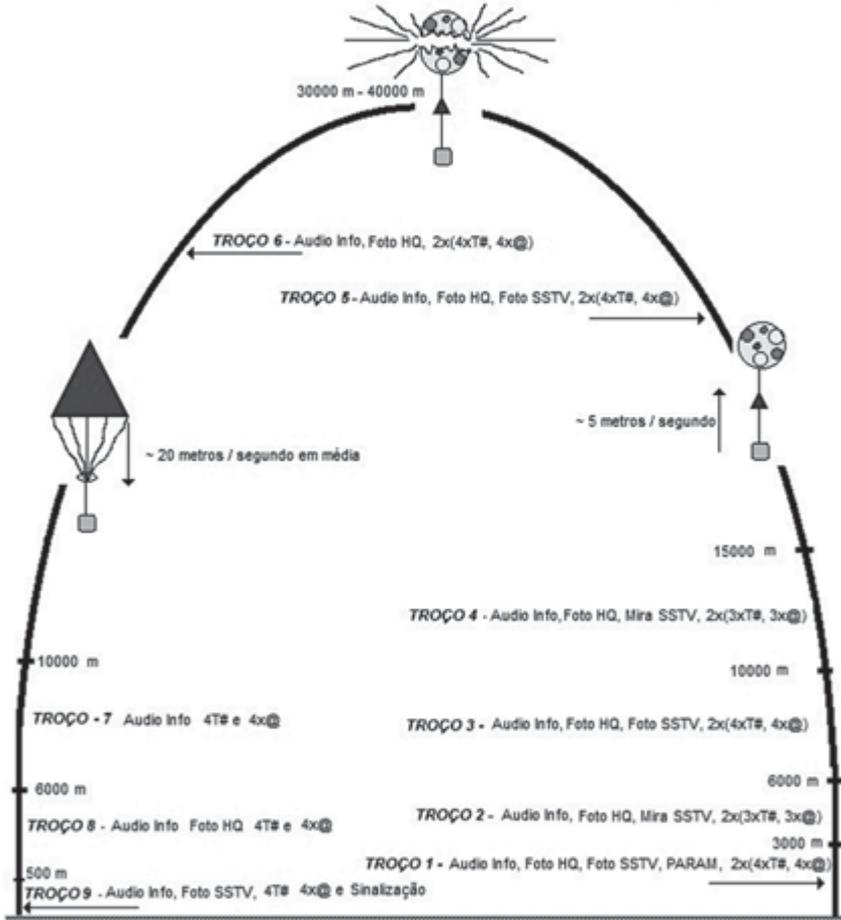


Foto HQ = Fotografia de Alta qualidade 2592x1944 pixels, guardada no software do Payload

Foto SSTV = Fotografia da terra enviada por SSTV em VHF

Audio Info = Informação de audio indicando o trecho de percurso e a direção do Payload

T# = Transmissão das Telemedidas em APRS/AX25

@ = Transmissão em APRS/AX25 das coordenadas obtidas do GPS

PARAM = Transmissão dos Parâmetros para a descodificação das Equações da Telemetria pelo protocolo APRS

Sinalização = Emissão de um alarme em audio e iluminação intermitente, para localização

Ex: Audio Info, Foto HQ, 2x(4xT#, 4x@) = Transmissão da informação de audio, Recolha de uma Foto de qualidade para o arquivo e transmissão de 4 Telemedidas e 4 Coordenadas por duas vezes

**Figura 5 - Plano de voo e de transmissões do Pay Load**

Os valores destas telemedidas são descodificados na recepção do Aprs.fi pelo protocolo APRS desde que esteja activada a função telemidada.

## 4 - PLANO DE VOO E TRANSMISSÕES

O plano de voo e de transmissões do balão é o mostrado na Figura 5 onde se vê que dividimos o percurso entre 9 troços situados entre altitudes definidas e dentro desse espaço a electrónica procede a determinadas medidas e transmissões.

*Exemplificando:*

**No Troço 5 transmite-se:**

Audio Info : Transmite-se em fonia a altitude e a direcção do balão (a subir ou a descer)

Foto HQ : Tira uma foto de Alta qualidade.

Foto SSTV transmite logo a foto em directo para SSTV Martin 1

2 x (4xT#, 4x@) : Transmite 4 vezes as telemedidas e 4 vezes as coordenadas e, este conjunto é transmitido duas vezes.

A repetição das transmissões tentam prevenir e existencia de erros causados por interferencias má propagação etc., porque um erro que haja, os dados são logo descartadas pelo corrector de erros Read Solom Code da recepção.

## 5 - O SOFTWARE

São 5 programas independentes que realizam todas as tarefas anteriormente indicadas; 4 deles são comandados pelo programa principal aprsstv.

Os programas foram escritos em linguagem C++ usando com o Visual Studio 2008 e um interface VisualGDB da SYSPROGS que deve ser baixado da net e que permite o Visual Studio faça uma compilação cruzada entre a aplicação em linguagem C no PC e o Raspberry PI em Linux, sendo que, o compilador, corre no PC.

**Ao** *iniciar o Visual Studio, na janela que aparece do Visual GDB basta entrar no Wizard LINUX para iniciar a programação. Para estabelecer*

**a comunicação entre o PC e o RASPI usei o programa PUTTY com uma ligação SSH através de um cabo de RJ 45.**

Através desta ligação SSH podemos usar um outro programa útil - WinSCP - que nos permite a gestão das pastas e dos Ficheiros dentro do RASPI a partir do nosso PC.

**Alguns** *programas escritos no Visual Studio e enviados por SSH para o Raspberry PI necessitam de aplicativos dentro Raspberry PI, obtidos por download directo. Estes programas requerem ainda mais uma recompilação em LINUX realizada na linha de comandos do Raspberry PI.*

*Por exemplo:*

Uma das funções do programa atctp.cpp é comunicar em RS232 com a placa do GPS por isso, o Raspberry PI vai necessitar de um aplicativo próprio para o efeito.

Assim, depois de se escrever e compilar o programa atctp.cpp no Visual Studio e enviá-lo pela ligação SSH para o RASPI, é necessário baixar o aplicativo rs232.c da net e instalá-lo no Raspberry PI.

Feita a instalação do rs232.c terá de ser feita a recompilação do programa atctp.cpp dentro do Raspberry PI na linha de comandos com seguinte comando:

```
g++ atctp.cpp rs232.c -Wall -o2 -o /home/pi/atctp/Release/atctp.
```

Só assim é que a comunicação RS232 do programa atctp.cpp funcionará!

### Programa 1: aprsstv.cpp

É o programa geral de transmissões em FM que chama e comanda os programas atctp.

cpp - atsstv.cpp - atequa.cpp - para a execução de tarefas específicas de cada um deles fazendo ciclos com a transmissão de dados em APRS nos 144,800 MHz, a transmissão de SSTV nos 144.430 MHz e a Fonia em 144.550 MHz terminando com um Reboot, Tipo Watch dog para recomeçar novamente .

## Programa 2: atctp.cpp

Programa que captura as medidas e armazena-as no Buffer de transmissão do seguinte modo:

- Aquisição de dados do GPS no formato NMEA GGA através da porta RS232
- Aquisição do valor da pressão atmosférica e temperatura interior por I2C com o ADC MCP3008
- Calcula a altitude em função da pressão atmosférica.
- Aquisição da temperatura exterior do Termistor via SPI com IC MCP3008
- Colocação dos dados telemétricos hexadecimal nos Buffers : tempintbuf, tempextbuf, pressaobuf, altitudetempbuf, velocidvertbuf
- Cria o formato APRS para transmissão da telemetria.
- Cria o “frame” AX25 e insere na trama os dados a transmitir.
- Seleccionar o ficheiro de áudio da pasta de ficheiros áudio pré-gravados, destinado à informação da altitude e da direcção que o balão segue.

## Programa 3: atequa.cpp

Cria o formato das equações de cada medida para a transmissão em APRS.

Cria os 4 “frames” AX25 das: MSG, PARAM, UNID, COEFS.

Cria os 4 ficheiros de áudio MSG.wav, PARAM.wav, UNID.wav, COEFS.wav para a emissão de APRS em FM.

## Programa 4: atfmtxctp.cpp

Programa que faz a transmissão em APRS

nos 144.800 MHz das Coordenadas, das Temperaturas, Pressão atmosférica e da Radiação Gama em formato de áudio preparados pelo programa atcp.cpp

## Programa 5: atsstv.cpp

Programa que formata num ficheiro de áudio a imagem retirada da camera para a transmissão da SSTV em Martin 1

## Programa 6: atfmtxstv.cpp

Programa que faz a transmissão em 144.430 MHz do audio da imagem SSTV criada pelo programa atsstv.cpp

**A** estrutura das transmissões durante do voo está dividida em 9 troços situados entre altitudes específicas.

**O** software verifica a altitude e dará as informações correspondentes ao troço onde o balão se encontra, como mostra a Figura 5.

Os colegas que queiram analisar e explorar o software com mais detalhe, podem descarregar deste site:

<http://www.ct4bb.com/aprstv.cpp>  
<http://www.ct4bb.com/atctp.cpp>  
<http://www.ct4bb.com/atfmtxstv.cpp>  
<http://www.ct4bb.com/atequa.cpp>  
<http://www.ct4bb.com/atfmtxctp.cpp>  
<http://www.ct4bb.com/atsstv.cpp>

O formato dos ficheiro está já com extensão .cpp (C plus plus ) para o Visual Studio.

Apenas para ler, e para quem não tem o Visual Studio, os programas com extensão .cpp podem ser abertos e lidos com o Notepad.

73 CT4BB  
(Não ao acordo ortográfico)