

# CanSat Júnior

4ª Edição

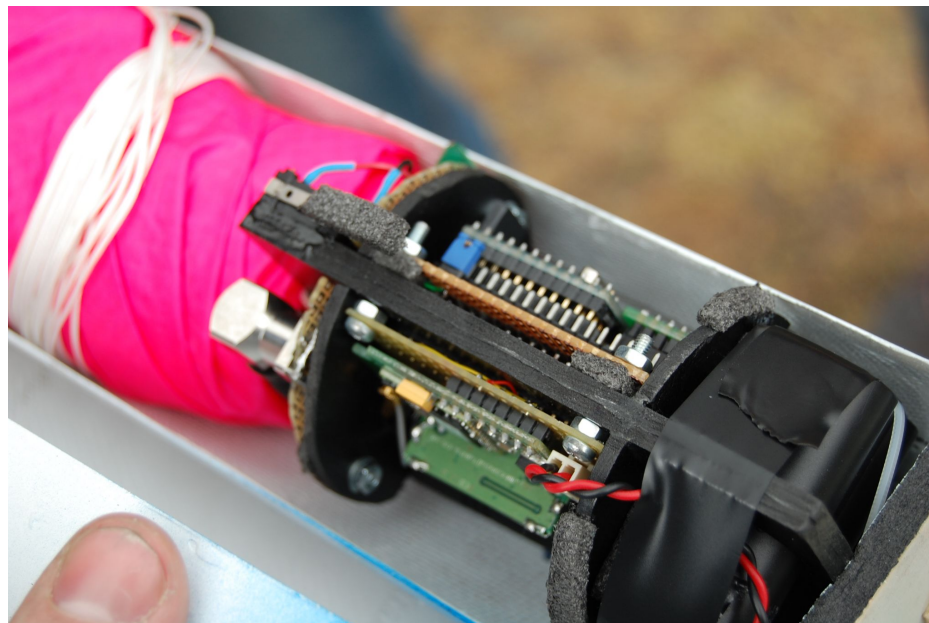
Workshop de Professores



**CANSAT JUNIOR**

# Can-quê?

- Simulação de satélite real no volume de uma lata de refrigerante
  - Lançado de até 1km de altura
  - Realiza missão científica
  - Aterra com segurança
- Desafios
  - Dimensão
  - Massa
  - Autonomia
  - Alcance de Transmissão
  - Fiabilidade
  - Resiliência
  - Recuperação

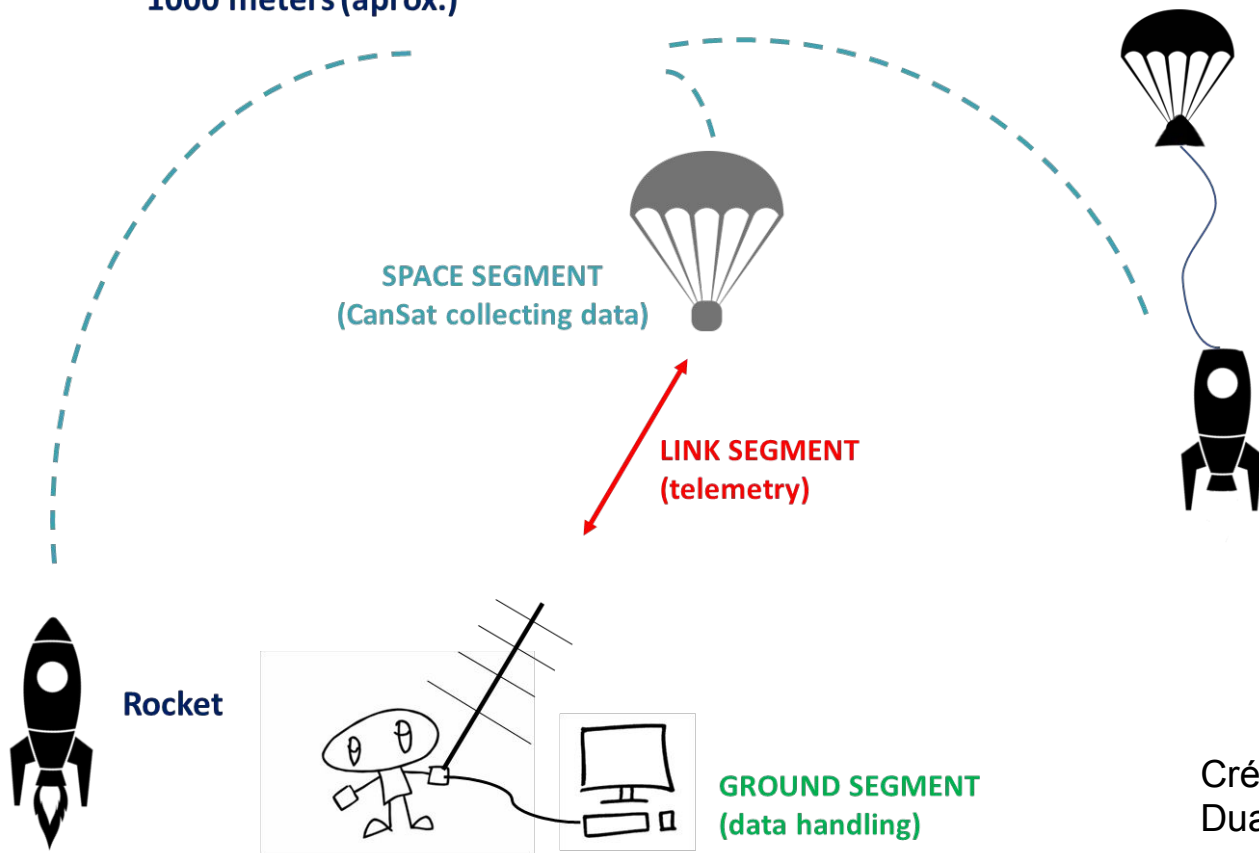


# Componentes típicas de um projeto CanSat

- **CanSat**
  - Sensores
  - Microcontrolador (Arduino)
  - Transceiver
  - Paraquedas
  - Invólucro
- **Estação Base**
  - Transceiver
  - Computador
  - Programa que guarde as transmissões do CanSat
  - Antena
- **Documentação**
  - **Checklists**
  - **Relatórios (pedidos e de testes)**
  - **Manuais**



1000 meters (aprox.)



Créditos da imagem:  
Duarte Cota

## CanSat Júnior

- Kit traz material necessário para desenvolver todo o projeto: construir antena, paraquedas e o CanSat;
- Não há “Missão Secundária” obrigatória;
- Construção simples;
- Sem necessidade de soldar;
- Programação simplificada (por blocos) com Scratch

Foco na aprendizagem e no método científico.

Desenvolvimento de competências úteis para os alunos.

# Divisão Temática

## Sistema de Recuperação

- Forças que atuam no satélite
- Atrito
- Velocidade vs aceleração
- Ilusão de subida do paraquedas

## Comunicações

- Ondas eletromagnéticas vs ondas mecânicas
- Modulação de sinal (AM/FM/...)
- Ruído e colisões de sinal
- Antenas

## Eletrónica e Programação

- Circuitos elétricos
- Estrutura de um robot (sensores, atuadores)
- Programação

## Sistema de Recuperação

- Forças que atuam no satélite
- Atrito
- Velocidade vs aceleração
- Ilusão de subida do paraquedas

## Comunicações

- Ondas eletromagnéticas vs ondas mecânicas
- Modulação de sinal (AM/FM/...)
- Ruído e colisões de sinal
- Antenas

## Eletrónica e Programação

- Circuitos elétricos
- Estrutura de um robot (sensores, atuadores)
- Programação

# Paraquedas

- Objetivo: velocidade terminal entre 8 e 11 m/s
- Tipos
  - Parapente permite controlo da descida - difícil de implementar
  - *Flat Shape* - fáceis de construir, conseguem pouco atrito
- Forma de um paraquedas *Flat Shape*
  - Mais lados = mais estabilidade
  - Incluir chaminé (*spill hole*) para estabilidade
- Testar (se possível gravar um vídeo)
- Recursos úteis:
  - <https://wiki.mchobby.be/index.php?title=ENG-CANSAT-PARACHUTE>



## Dimensionar um paraquedas

No paraquedas temos 2 forças a atuar na vertical: gravidade e atrito

Resolver  $F_D = F_G$  na velocidade terminal

Força de atrito:  $F_D = \frac{1}{2} \rho C_D A v^2$

Força da gravidade:  $F_G = m g$

$m$  - massa do satélite (kg)

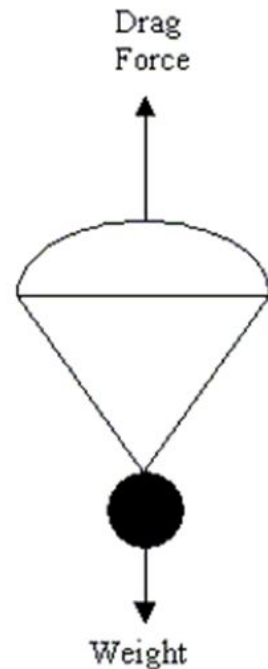
$g$  - aceleração gravítica na terra ( $m/s^2$ )

$\rho$  - densidade do ar ( $kg/m^3$ )

$C_D$  - **coeficiente de atrito - obtido experimentalmente** (0.75-0.8 para círculo)

$A$  - área do paraquedas ( $m^2$ ) - valor a obter

$v$  - velocidade terminal (m/s)



# Comportamento desejado do paraquedas

1. Assim que o CanSat é ejetado, o paraquedas deve colocar-se no sítio certo naturalmente.
  - Dobrar o paraquedas corretamente
  - Criar dispositivo de abertura se considerarem necessário
2. A queda deve ser sem o CanSat a baloiçar ou a rodar sobre si mesmo.
  - Perigo de enrolamento dos fios - usar destorcedor
  - Chaminé
3. Para paraquedas circular o diâmetro não deve passar dos 35 cm.

# Recomendações

- Dimensão dos “fios”:
  - Maior ou igual ao diâmetro do paraquedas
  
- Dimensão da chaminé
  - 0,3 vezes diâmetro do paraquedas

# Vamos lançar?



## Sistema de Recuperação

- Forças que atuam no satélite
- Atrito
- Velocidade vs aceleração
- Ilusão de subida do paraquedas

## Comunicações

- Ondas eletromagnéticas vs ondas mecânicas
- Modulação de sinal (AM/FM/...)
- Ruído e colisões de sinal
- Antenas

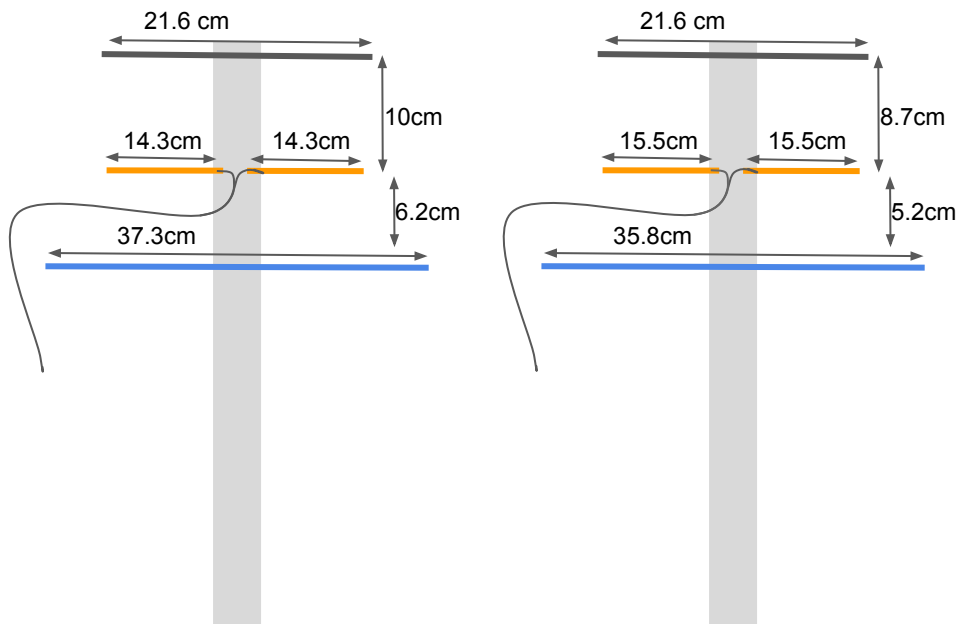
## Eletrónica e Programação

- Circuitos elétricos
- Estrutura de um robot (sensores, atuadores)
- Programação

# Antena

- Para o CanSat: fio monopolo de quarto de comprimento de onda ( $\lambda \approx 0.69\text{m}$ )
  - Dimensionamento e construção fáceis
  - Omnidirecional (perpendicular à antena)
  - Vem cortada e soldada no kit
- Para a estação terrestre: Yagi-Uda
  - Dimensionamento e construção complicados - partir de um modelo já existente e ajustar
  - Direcional
  - Material necessário à construção incluído no kit
- Devem conseguir um alcance de 1km (sem obstáculos)

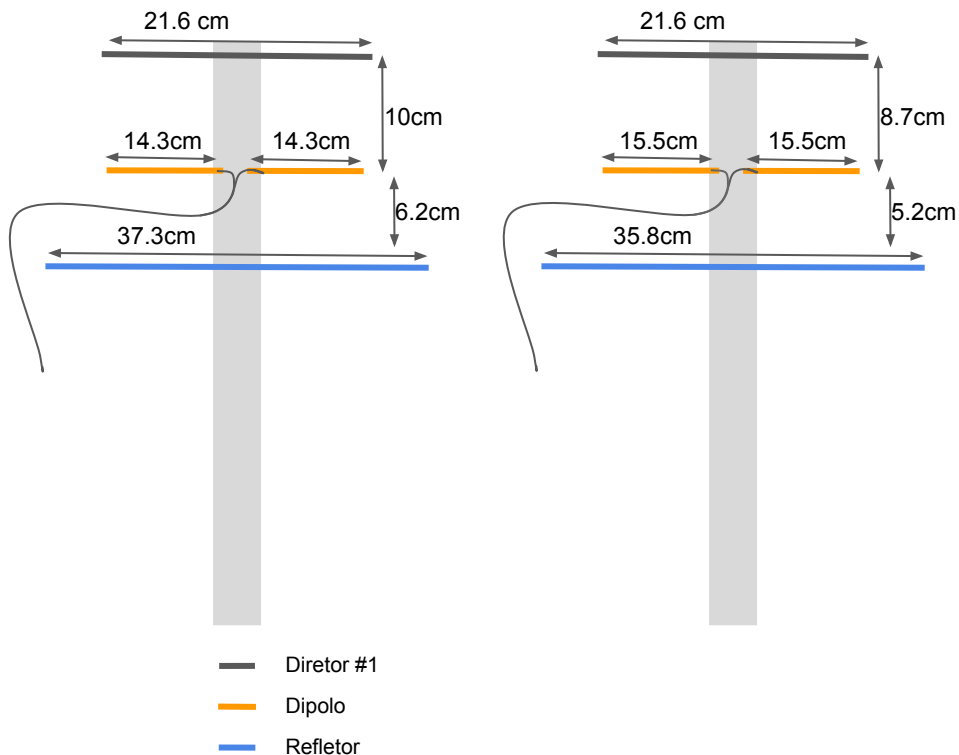
# Antena Yagi-Uda



- Diretor #1
- Dipolo
- Refletor

- A calha incluída no kit serve de *boom* (eixo)
- A fita métrica (também incluída no kit), cortada forma os elementos (refletor, dipolo e diretor/diretores) da antena, transversais ao *boom*

# Antena Yagi-Uda



- Calculadora de dimensões:

[https://www.changpuak.ch/electronics/yagi\\_uda\\_antenna\\_DL6WU.php](https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php)

<https://www.rfwireless-world.com/calculators/3-element-Yagi-Antenna-Calculator.html>

Reflector length =  $0.495 \cdot \lambda$

Dipole length =  $0.473 \cdot \lambda$

Director length =  $0.440 \cdot \lambda$

Reflector to Dipole spacing =  $0.125 \cdot \lambda$

Dipole to Director spacing =  $0.125 \cdot \lambda$

$\lambda = c/f$

$\lambda$ -Wavelength in meters

c-Velocity of propagation in air ( $3 \cdot 10^8$  m/s)

f-Carrier frequency in MHz



## Transceivers

- RFM69HCW - consegue sintetizar frequências entre os 424 MHz e 510 MHz
- Código fornecido/gerado opera-o em modo-pacote, mas simula transmissão contínua
  - Alguns dados podem ficar por enviar até o início da próxima transmissão
  - A operação *flush* força o envio destes dados pendentes
- Se forem detetados erros de transmissão em parte dos dados (CRC16), essa parte é rejeitada
- Recomendação: dados em formato CSV
  - Uma transmissão por linha
  - Campos/colunas separados(as) por ponto e vírgula (;)

# Mãos à obra



## Sistema de Recuperação

- Forças que atuam no satélite
- Atrito
- Velocidade vs aceleração
- Ilusão de subida do paraquedas

## Comunicações

- Ondas eletromagnéticas vs ondas mecânicas
- Modulação de sinal (AM/FM/...)
- Ruído e colisões de sinal
- Antenas

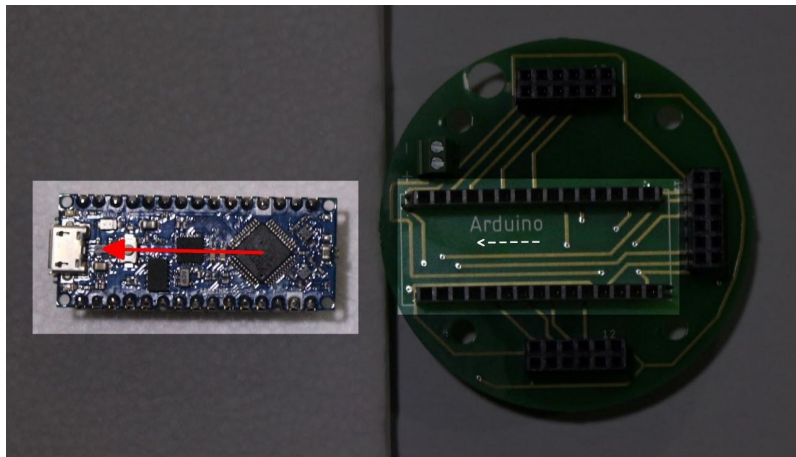
## Eletrónica e Programação

- Circuitos elétricos
- Estrutura de um robot (sensores, atuadores)
- Programação

## O Kit: Conteúdos

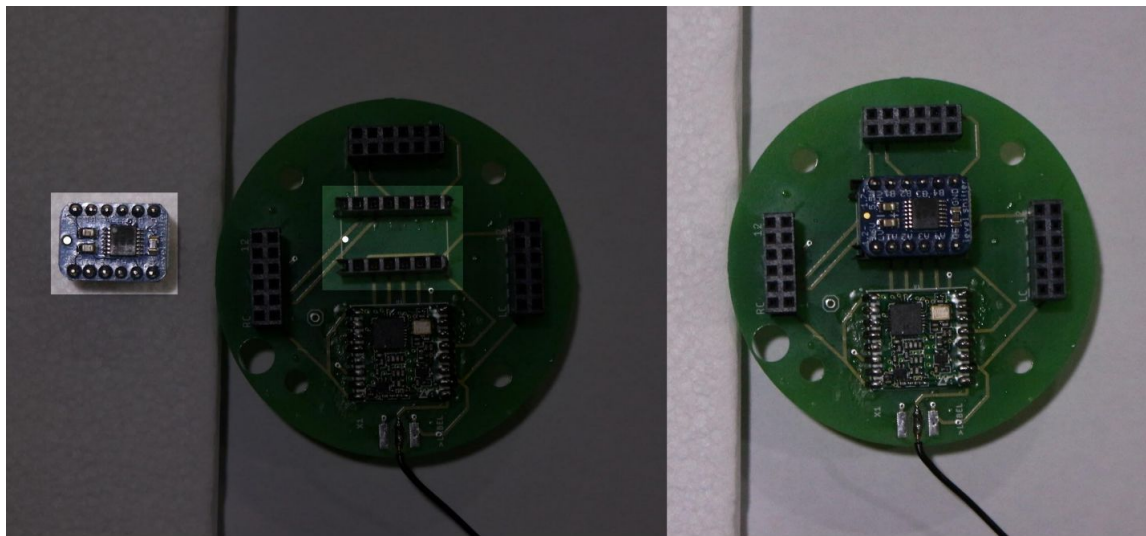
Para além dos materiais necessários para o paraquedas e antena, o kit contém várias PCBs e componentes:

- PCBs para microcontrolador (x2)
- Microcontrolador (Arduino Nano) (x2)



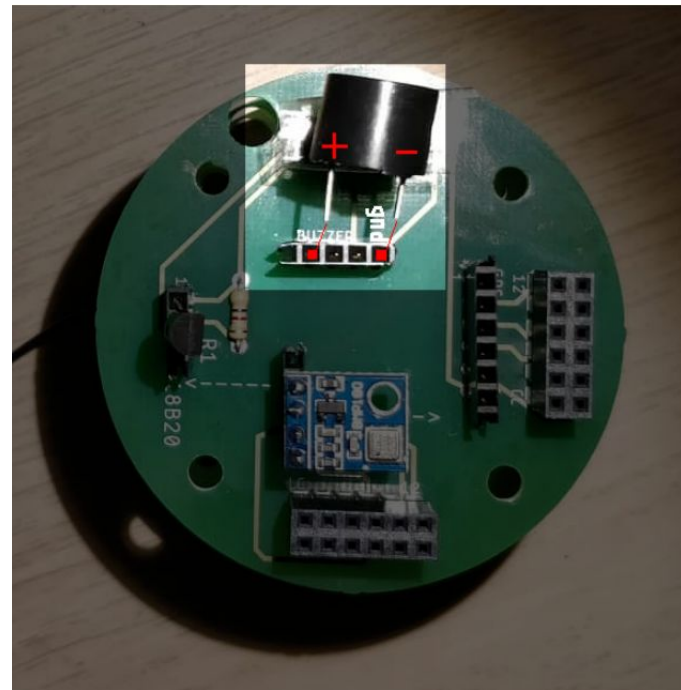
## O Kit: Conteúdos

- PCBs para *transceiver* (x2)
- *Transceiver* (RFM69HCW) (x2) (soldados nas placas respetivas)
- *Level Shifter* (TXB0104, componente necessário à comunicação) (x2)



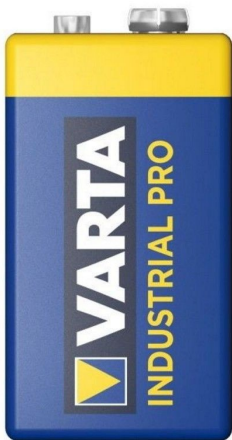
## O Kit: Conteúdos

- PCB para sensores (x1)
- Sensor de Temperatura (DS18B20)
- Sensor de Pressão (BMP180)
- *Buzzer* ativo



## O Kit: Conteúdos

- Componentes estruturais (invólucro, tampa, porcas, anilhas, etc.)
- 2 pilhas de 9V



## O Kit

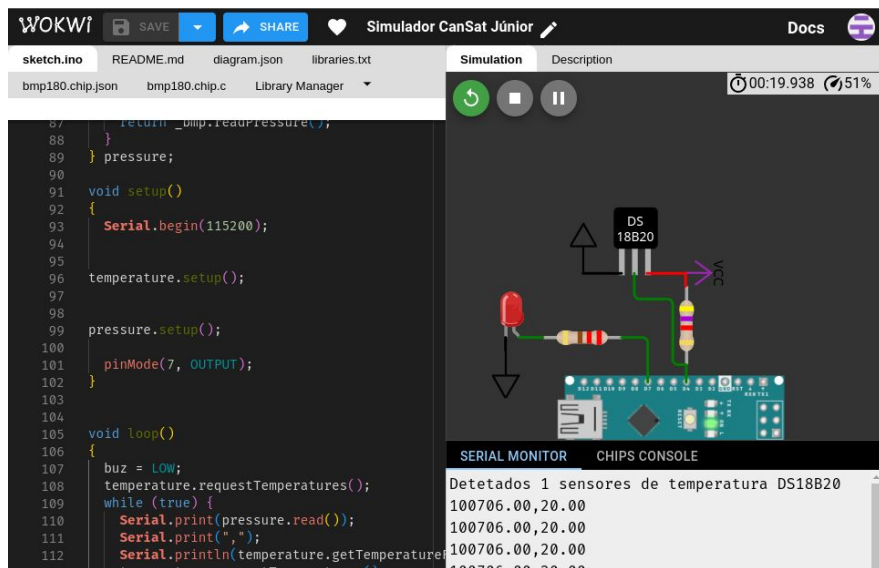
- Manual completo em <https://cj.breda.pt/>
- Quase todos os componentes podem ser facilmente substituídos/reutilizados
- Estação base usa *Arduino* e *Transceiver*
- Cuidado a montar!
  - Atenção às indicações de orientação dos componentes (verificar o manual)
  - Recomenda-se criar separação entre as várias placas
  - Desafia-se as várias equipas a escolherem o melhor posicionamento para o *master switch*
  
- Guardem uma pilha para a competição!



# Programação

- Um Arduino é tipicamente programado com C/C++
  - Erros de compilação ininteligíveis e tiros no pé frequentes
- Editor baseado em Scratch disponível em: <https://cj.breda.pt/>
  - Gera código C/C++ que terá de ser compilado pela IDE do Arduino
  - Disponibilizamos exemplos simples para cada sensor, rádio e buzzer
  - Todos os blocos estão documentados no manual
- **NOVO:** Simulador
  - Suporta sensores de pressão e temperatura
  - Buzzer simulado como LED
  - Rádio ainda não disponível

# Mãos à obra



The screenshot shows the WOKWI simulator interface. The top bar includes 'WOKWI', 'SAVE', 'SHARE', 'Simulador CanSat Júnior', and 'Docs'. Below the bar, there are tabs for 'sketch.ino', 'README.md', 'diagram.json', and 'libraries.txt'. The left pane shows the code editor with the following code:

```
87     return _bmp.readPressure();
88   }
89 } pressure;
90
91 void setup()
92 {
93   Serial.begin(115200);
94
95   temperature.setup();
96
97   pressure.setup();
98
99   pinMode(7, OUTPUT);
100 }
101
102 void loop()
103 {
104   buz = LOW;
105   temperature.requestTemperatures();
106   while (true) {
107     Serial.print(pressure.read());
108     Serial.print(",");
109     Serial.println(temperature.getTemperature());
110   }
111 }
```

The right pane shows the simulation window with a breadboard circuit. A red buzzer is connected to pin 7. A resistor is connected between pin 7 and ground. A DS18B20 sensor is connected to the breadboard. The simulation window includes a 'SERIAL MONITOR' and 'CHIPS CONSOLE' section. The SERIAL MONITOR displays the following output:

```
Detetados 1 sensores de temperatura DS18B20
100706.00,20.00
100706.00,20.00
100706.00,20.00
100706.00,20.00
```

# Programação

- Disponibilizamos código de apoio para a estação base
  - Programa a carregar no Arduino e programa de captura de dados para correr no computador
- Não esquecer: enviar tempo juntamente com os dados!
  - Ver bloco “Tempo Ativo” (função *millis* em C/C++ do Arduino)

# Recomendações finais

- Leiam manual e regulamento!
- Definem objetivos claros (em concordância com os regulamentados)
  - Para evitar problemas de falta de tempo, priorizem-nos
- Documentem tudo o que fazem (para que seja contabilizado)
- Documentem tudo o que fazem para **provar que cumpriram os vossos objetivos**
  - Procedimento de testes/checklist e resultados

# Recomendações finais

- Não tenham medo de errar!
  - Erros bem documentados (sintomas, causa, proposta de correção, lições aprendidas) podem ser mais valiosos do que um projeto a funcionar bem
- Deem-nos feedback e façam perguntas
  - O projeto é recente e tem margem para evoluir

## FAQ

- A temperatura medida é  $-127^{\circ}\text{C}$ . O que se passa?
  - Verificar ligação do sensor de temperatura;
  - ATENÇÃO: se estiver ligado ao contrário vai estar **MUITO QUENTE**.
- Erro de compilação “QualquerCoisa.h: No such file or directory”
  - Verifique que todas as bibliotecas indicadas no manual estão instaladas.
- Não consigo carregar o meu programa no Arduino: “Ocorreu um erro a enviar o rascunho”
  - Verifique a ligação do Arduino ao computador;
  - Garanta que o modelo da placa selecionado é o correto;
  - Garanta que a porta de série selecionada é a correta.

# Contactos

- Estamos disponíveis para responder quaisquer questões que surjam, bastam mandar um email!
  - ESERO: [esero@cienciaviva.pt](mailto:esero@cienciaviva.pt) (questões gerais, regulamento, logística, certificados, etc.)
  - Filipe Almeida : [filipedgeraldo@gmail.com](mailto:filipedgeraldo@gmail.com) (questões técnicas, foco em eletrónica)
  - André Breda: [cj@breda.pt](mailto:cj@breda.pt) (questões técnicas, foco em programação)