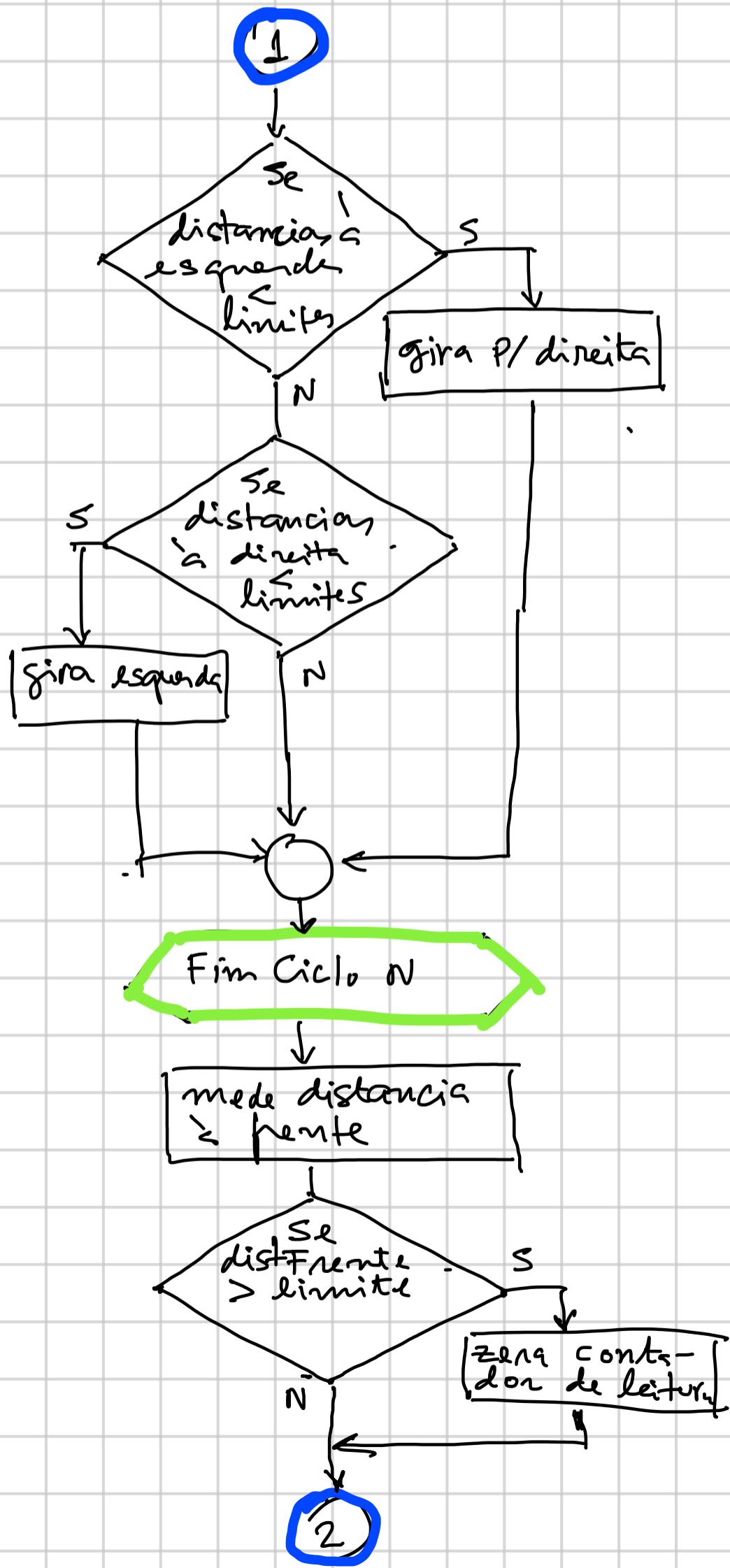
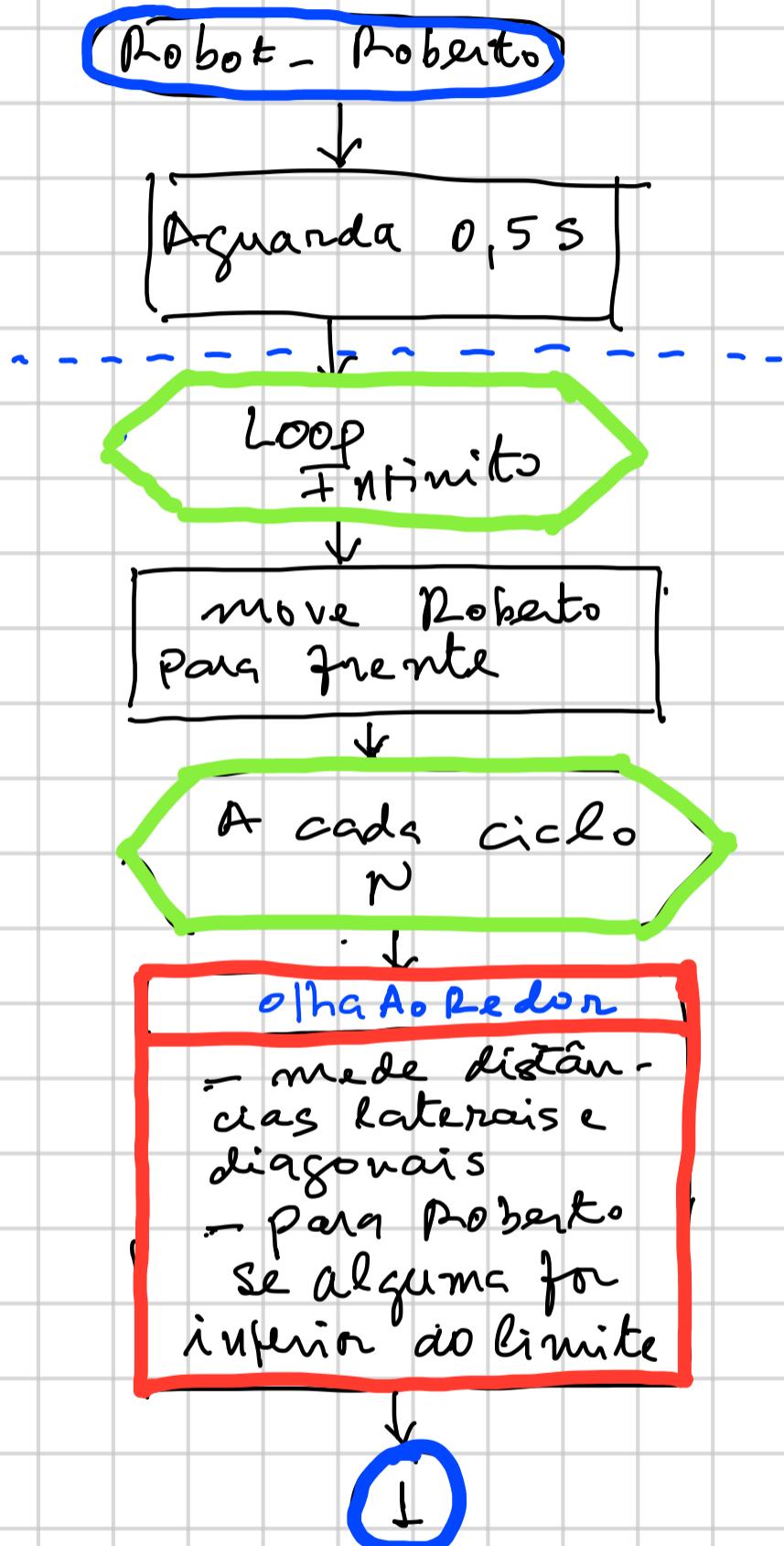
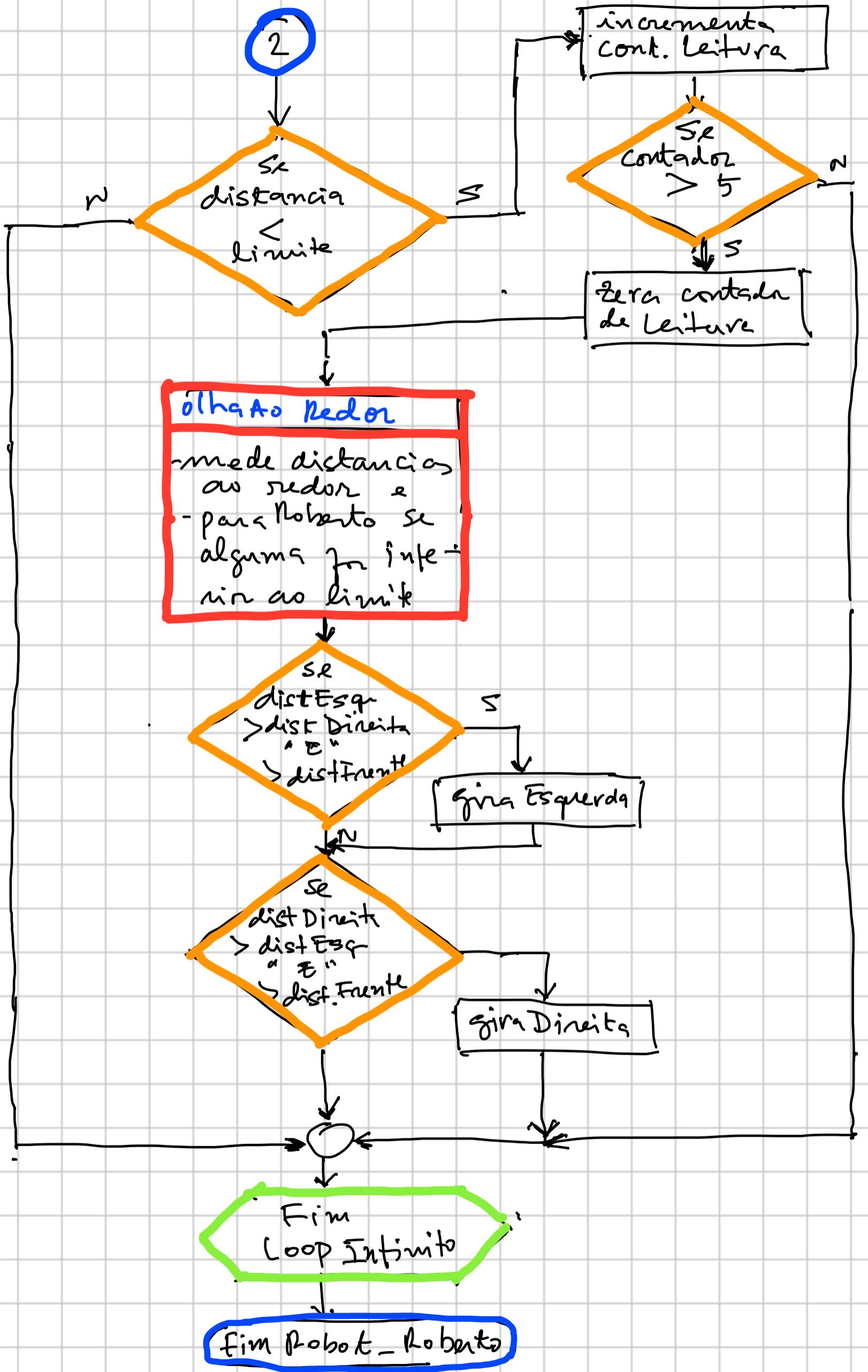


Projeto 7 - Robot\_Roberto (Ana Clara)  
22/10/2020 - Augusto Sherman

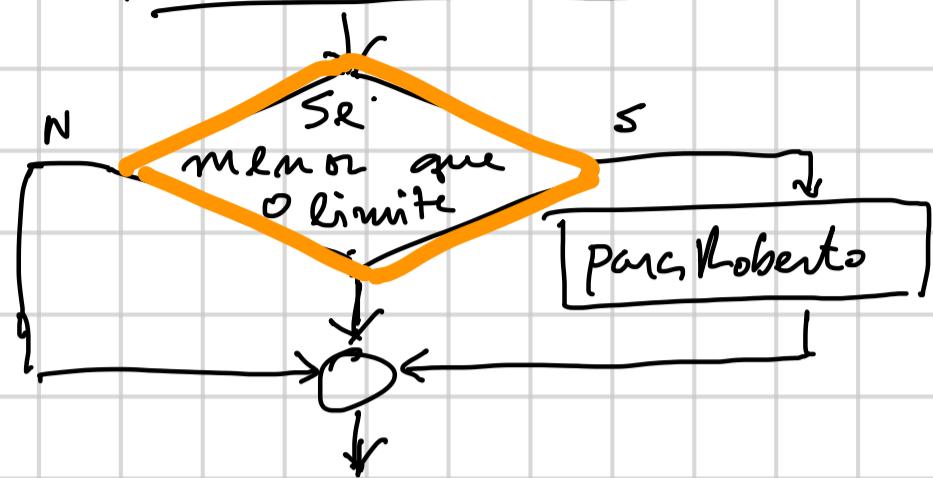
1- Fluxograma Principal



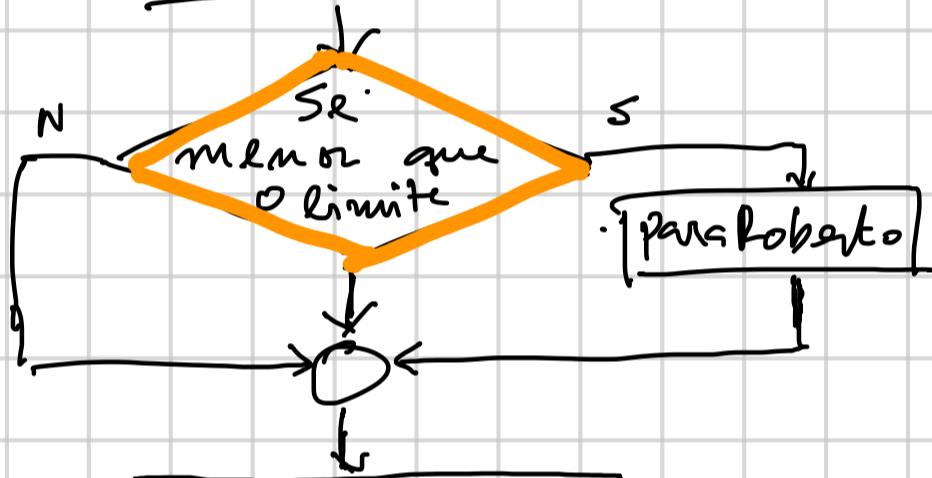


## Olha Ao Redor

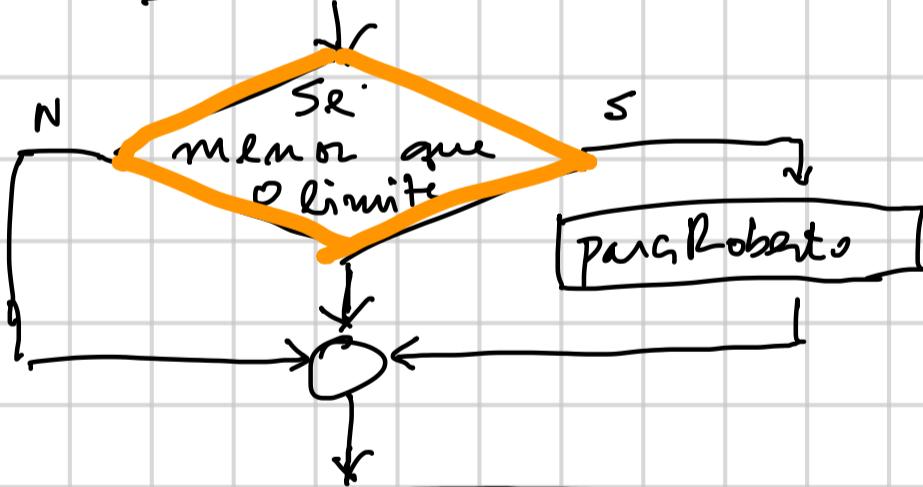
[le<sup>r</sup> dist Frente]



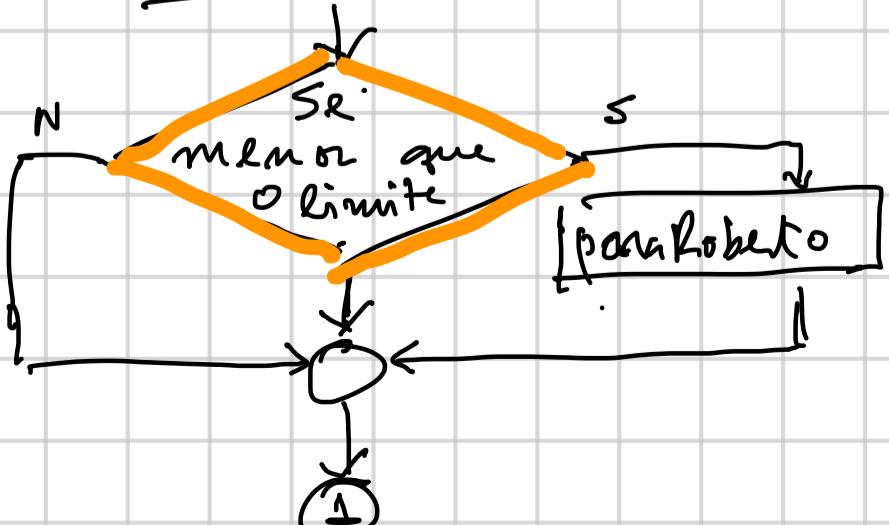
[le<sup>r</sup> dist Diag Esq]



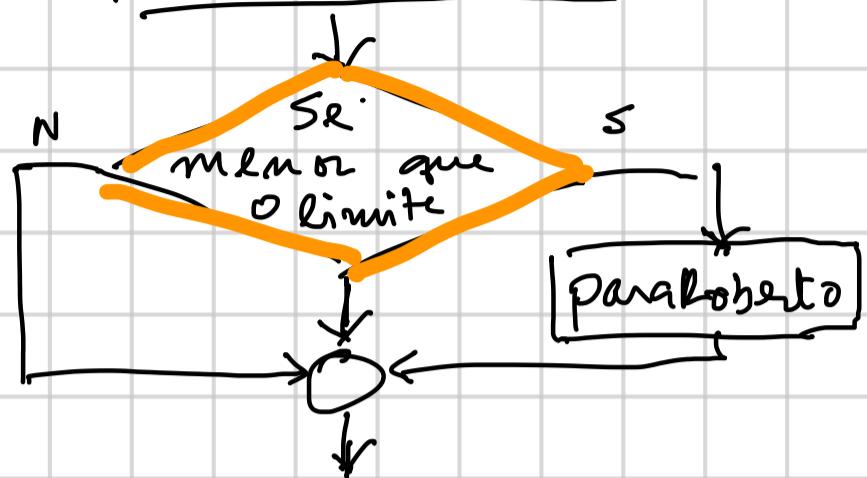
[le<sup>r</sup> dist Esquerda]



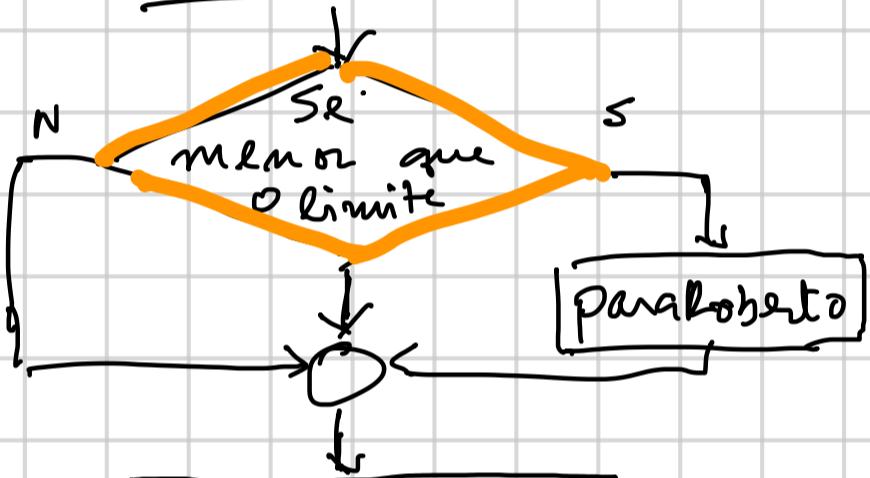
[le<sup>r</sup> dist Diag Freq]



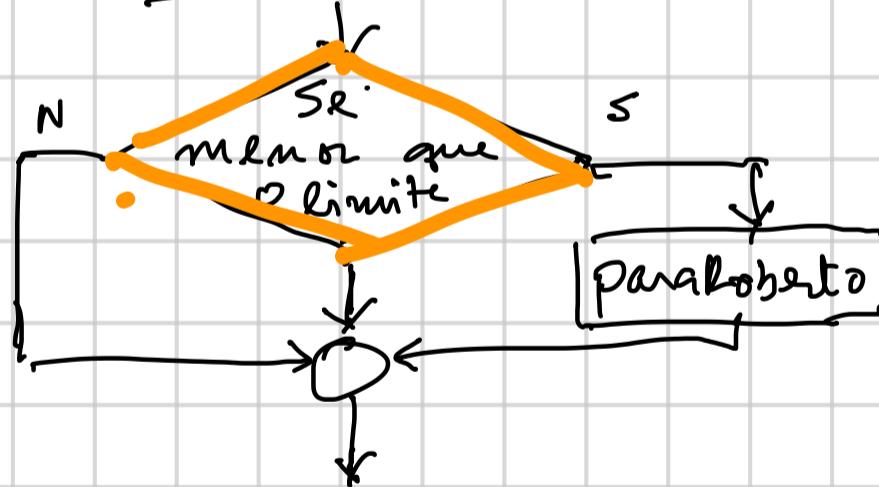
[le<sup>r</sup> dist Frente]



[le<sup>r</sup> dist Diag Direita]



[le<sup>r</sup> dist Direita]



Olha Para Frente

Fim Olha Lateral

## 2 - Funções

void moveRoberto.Frente()	✓	void msg.Susto()
void moveRoberto.Tros()	✓	void msg.BomDia()
void giraRoberto.Direita()	✓	
fireRoberto.Esquerda()	✓	
void ParaRoberto()	✓	
void olhaAoRedor()		
int LeDistancia()	✓	

## 3 - Classes

Cabeça (direção)

- frente, Diag Direita, Direita  
Diag Esquerda, Esquerda

## 4 - Variáveis

int distFronte  
 int distDireita  
 int distDiagDireita  
 int distDiagEsquerda  
 int distEsquerda

## 5 - Constantes

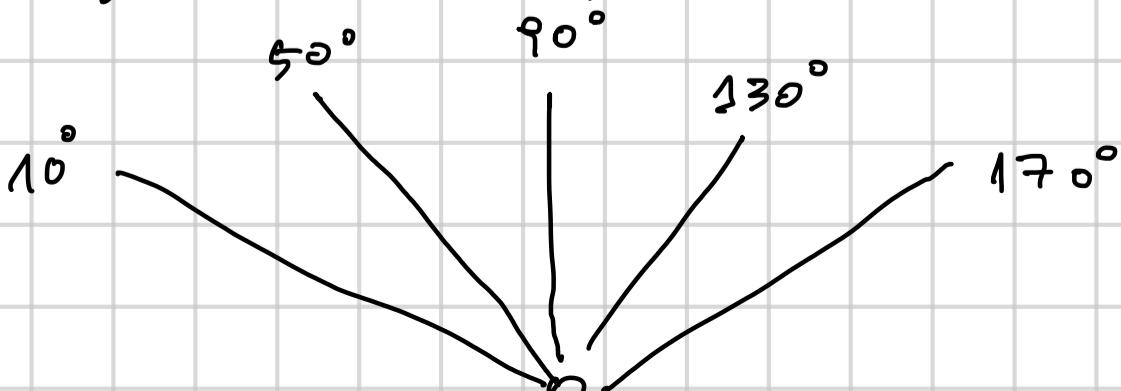
int limiteDistancia  
 int limiteDistLateral  
 int tempoDeRiro  
 int frente  
 int diagDireita  
 int direita  
 int diagEsquerda  
 int Esquerda  
 int velocidade

## C - Pinos do Arduino

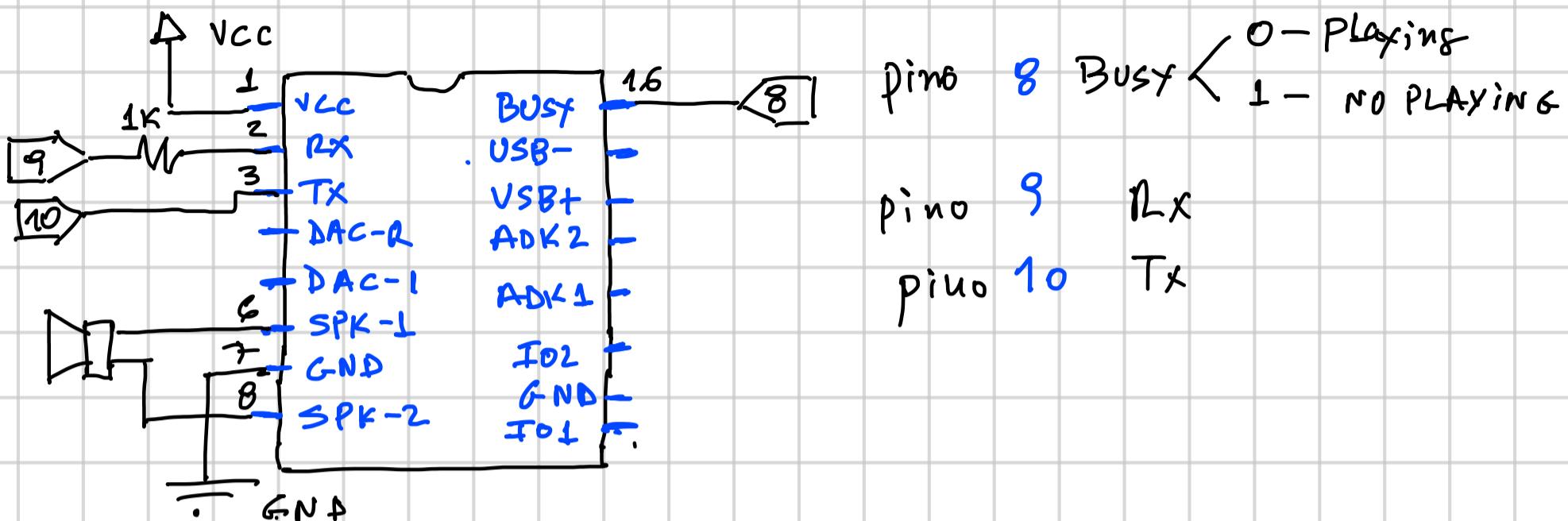
motor	mD veloc	- 5	Saídas PWM (6 e 11, até versão 4)
	mD pins < 2	- A4 e A5	
	mE pins < 2	- A2 e A3	
	mE veloc	- 6	
sensor distância	trigPin	- 3	Saída digital (13, versão 4)
	echoPin	- 2	Entrada digital (12, versão 4)
servo	velocPin	- 0	Entrada Analógica A0
	ServoPin	- 7	Saída PWM
mp3	dfp (Serial)	- 9 e 10	RX, TX
	dfp Busy	- 8	Entrada digital
IRremote	recepPin	- 4	Entrada
IR	sensIr Direito	- A1	
	sensIr Esquerdo	- A0	
Automático/ controle F2	modo Controle	- 11	0 → K → M → 1K → VCC
	Tx BLE	- A2	
Bluetooth HM-10	Rx BLE	- 13	

pinos disponíveis - 0, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,

- OBS.
- 1) a biblioteca <Servo> é incompatível com o uso dos pinos 9 e 10 em PWM
  - 2) a biblioteca <NewPing> é incompatível com a <IRremote>
  - 3) a biblioteca <IRremote> é incompatível com analogWrite() nos pinos 3 e 11
  - 4) Ângulos de medir giro de distância



# DFPlayer mini MP3



Pin 8 Busy  
0 - Playing  
1 - NO PLAYING

Pin 9 RX

Pin 10 TX

## 1) Bibliotecas

- <SoftwareSerial.h> - para comunicar com o DFPlayer
- <DFRobotDFPlayerMini.h> - para comandar o DFPlayer

## 2) Objetos

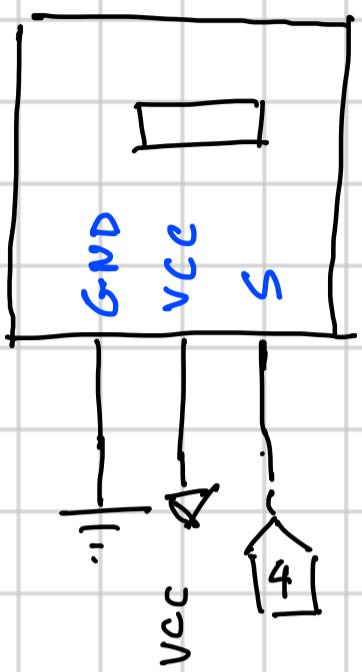
- SoftwareSerial dfpSerial(9,10) - cria a comunicação serial entre DFPlayer nos pinos 9 e 10
- DFRobotDFPlayerMini dfp - cria o "dfp"

## 3) Comandos

- dfpSerial.begin(9600) - iniciando a comunicação serial com o DFPlayer
- dfp.begin(dfp.Serial) - iniciaiza DFPlayer
- dfp.setTimeout(500) - time out do DFPlayer

- `dfp.volume(25)` - define volume do áudio do DFPlayer
- `dfp.EQ(DFPLAYER_EQ_NORMAL)` - define Equalizador normal para o DFPlayer
- `dfp.setOutputDevice(DFPLAYER_DEVICE_SD)` - define a leitura das músicas do Cartão SD
- `dfp.readFilesCountsInFolder(n)` - verifica o número de arquivos na pasta "n" do cartão SD
- `dfp.playFolder(n, msg-m)` - toca a mensagem "msg-m" da pasta "n".
- `dfp.playMp3Folder(n)` - toca a música n da pasta "mp3"
- `dfp.advertise(n)` - toca a faixa n da pasta "advert".

## Receptor do Controle de IR



### Comandos

- ✓ \* - modo automático / controle IR
- ✓ 0 - ON/OFF
- ✓ # - liga/desliga música
- ✓ ↑ - para frente
- ✓ ↓ - para trás
- ✓ → - para direita
- ✓ ← - para esquerda
- ✓ OK - parado
- ✓ 1 - Volume -
- ✓ 2 + <1><2> - música <1><2>
- ✓ 3 - Volume +
- ✓ 4 - Prv music
- ✓ 5 - Pausa music
- ✓ 6 - next music
- ✓ 7 - msg -
- ✓ 8 - MENU
- ✓ 9 - msg +

OBS: 1) Inicia no modo automático e LIGADO

### 1) Biblioteca

- <`IRremote.h`> - para comunicar com o receptor de IR

### 2) Objeto

- `IRrecv` receptor (receptorPin) - cria o "receptor"

### 3) Estructura

- decode - result comando — cria a estrutura de dados "comando".
  - decode-type
  - value
  - bits
  - rawbuf [3]
  - rawlen
- protocolo (ex. NEC...)
- Código recebido
- nº bits recebido
- timing bruto
- nº de registros em rawbuf [ ]

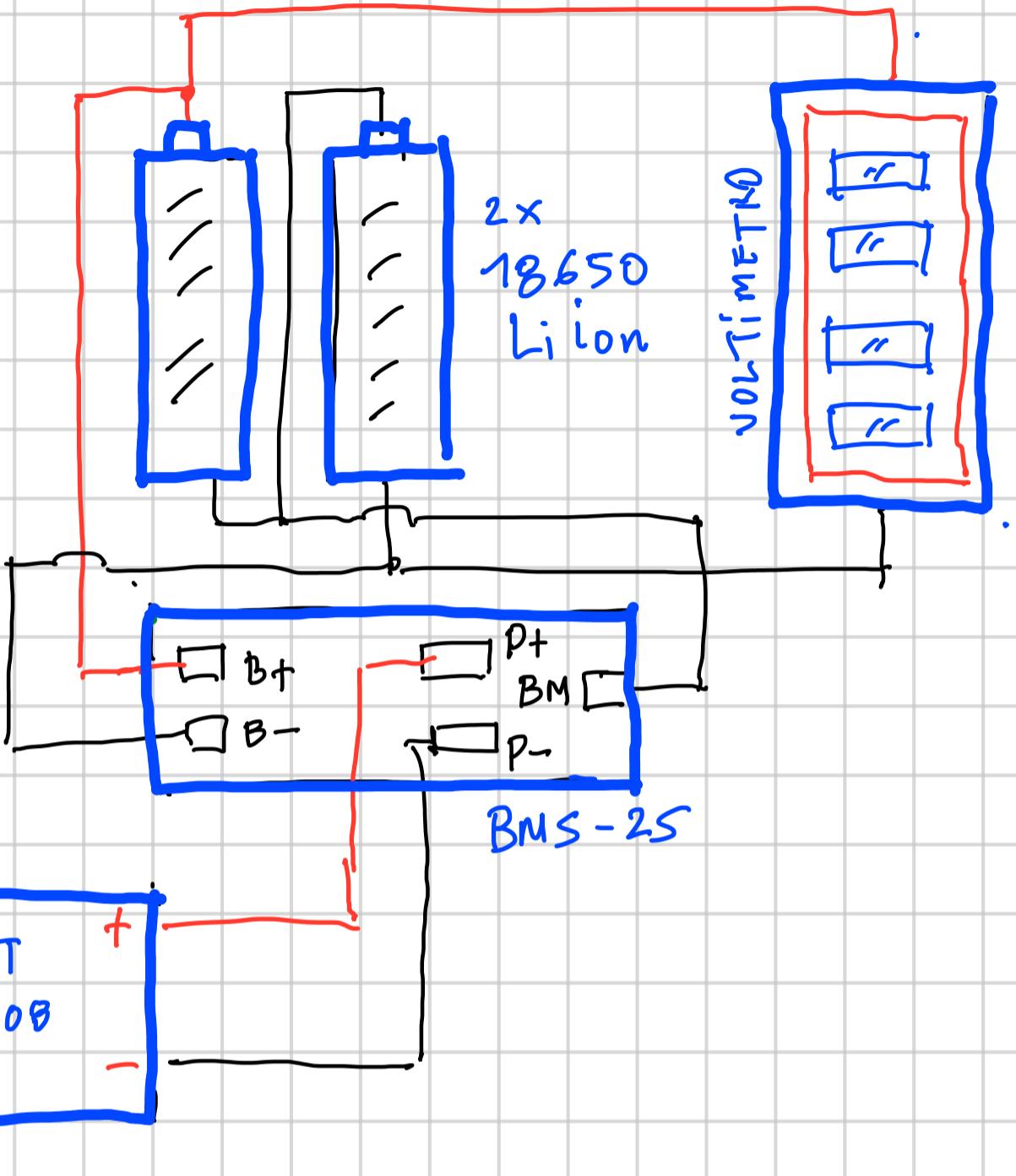
### 4) Comandos

- int receptor.decode (& comando) — retorna "1" se recebido um código, "0" se não. os dados recebidos estão na estrutura "comando"
- receptor.resume () — prepara para receber novo comando.
- receptor.enableIR () — inicializa o receptor
- receptor.blink13 (true) — pulsa led 13 ao receber sinal de IR
- receptor.comando.overflow () — 'true' se ocorrer

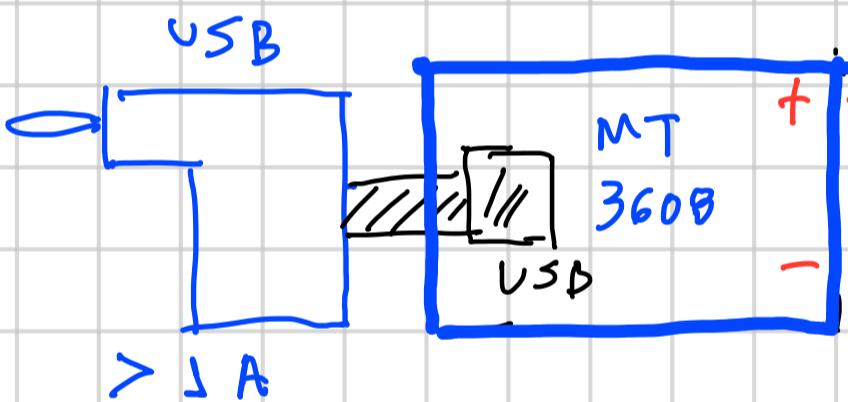
# Carregador de Baterias - Direto

## Material

2x 18650  
1x MT3608 USB  
1x BMS-2S  
1x VOLTMETRO  
1x CARREGADOR USB  
1x Suporte 2 bat 18650



## CARREGADOR



OBS. - As baterias podem ser carregadas sem retirá-las do suporte contido no Roberto.

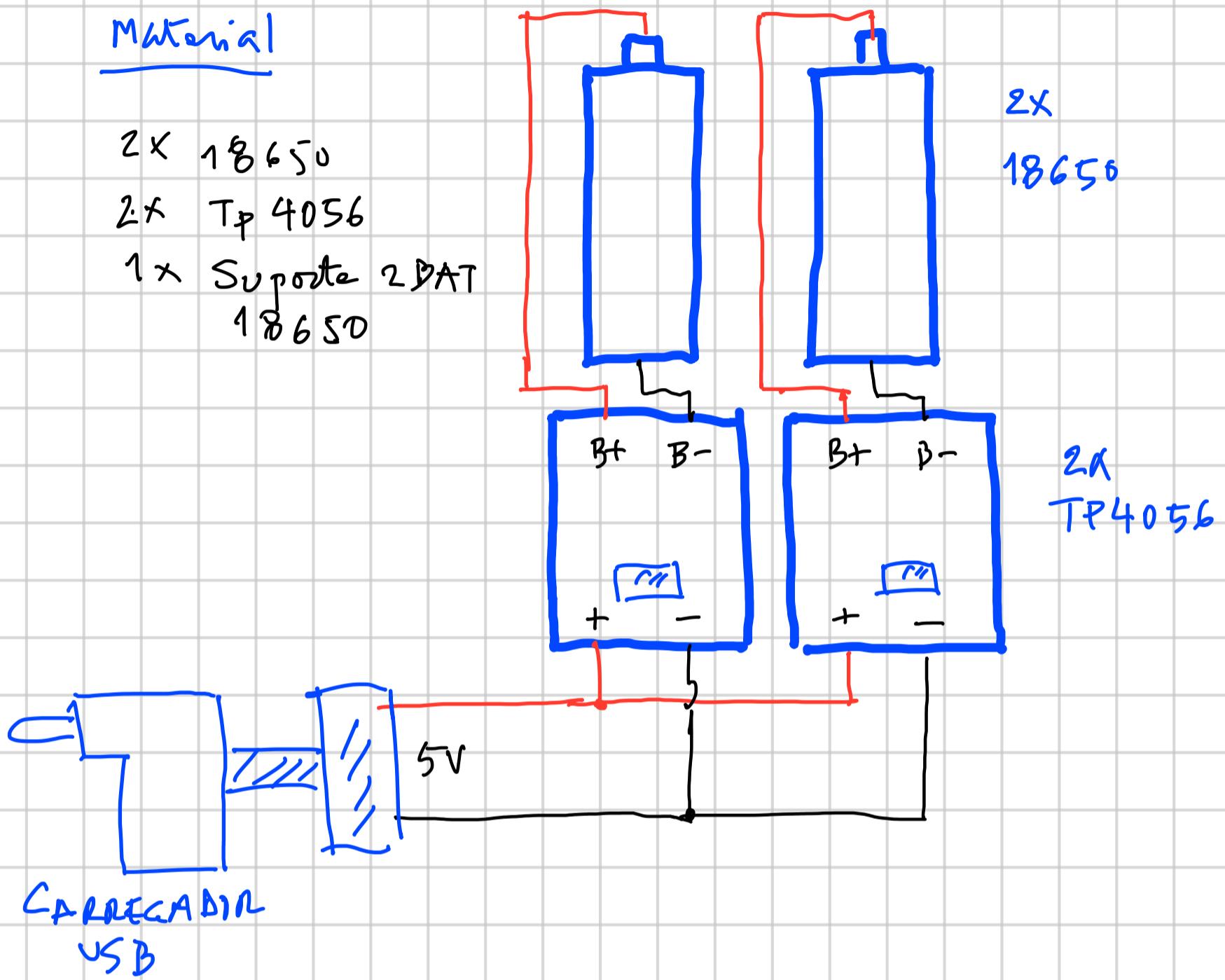
# Carregador de Baterias - Indireto

Material

2x 18650

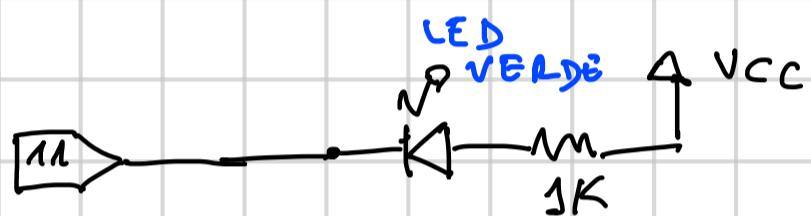
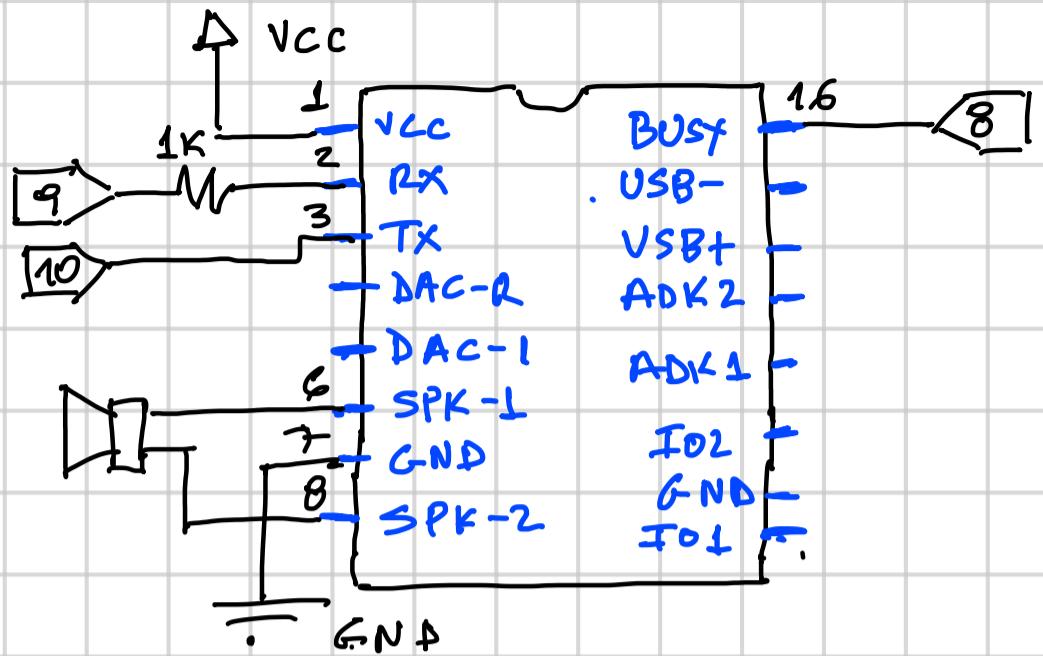
2x TP4056

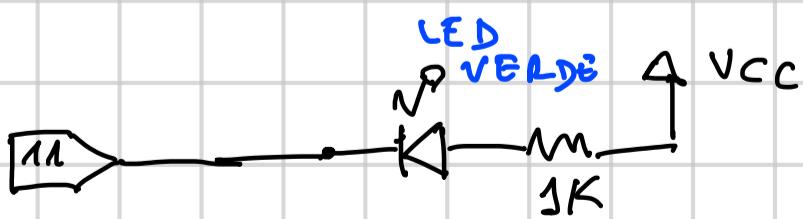
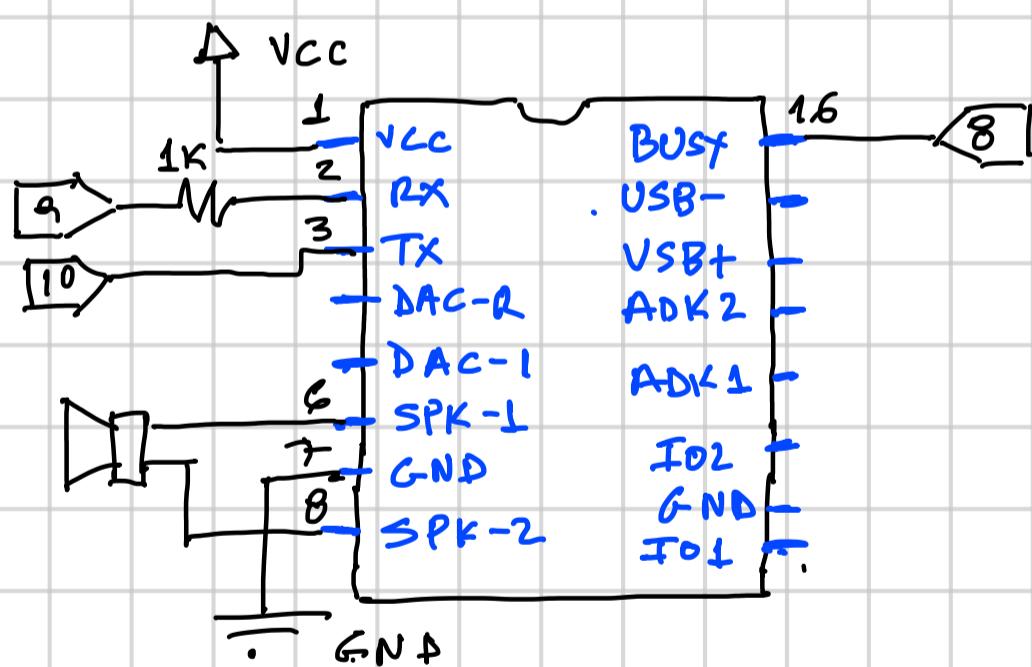
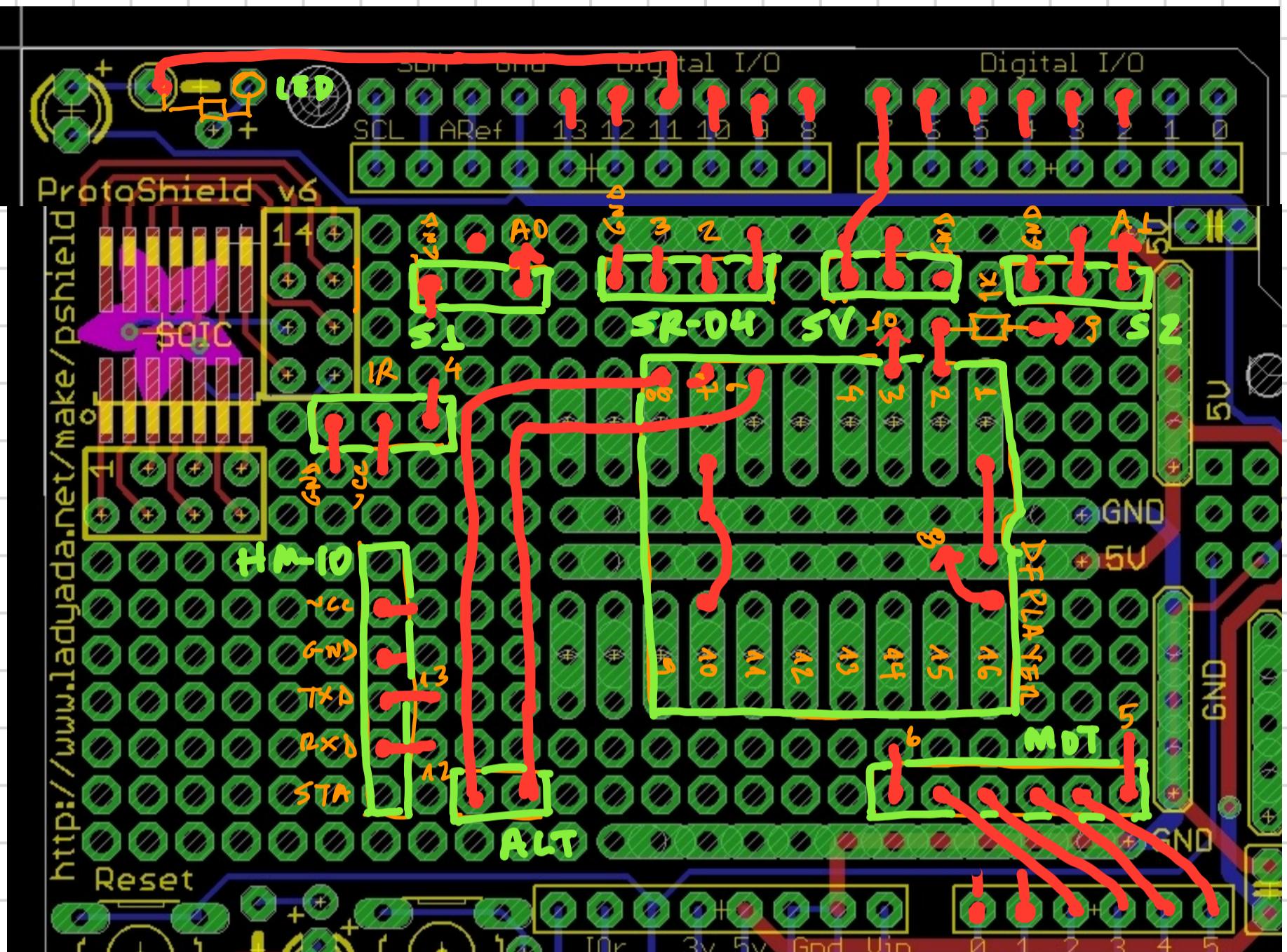
1x Suporte 2BAT  
18650

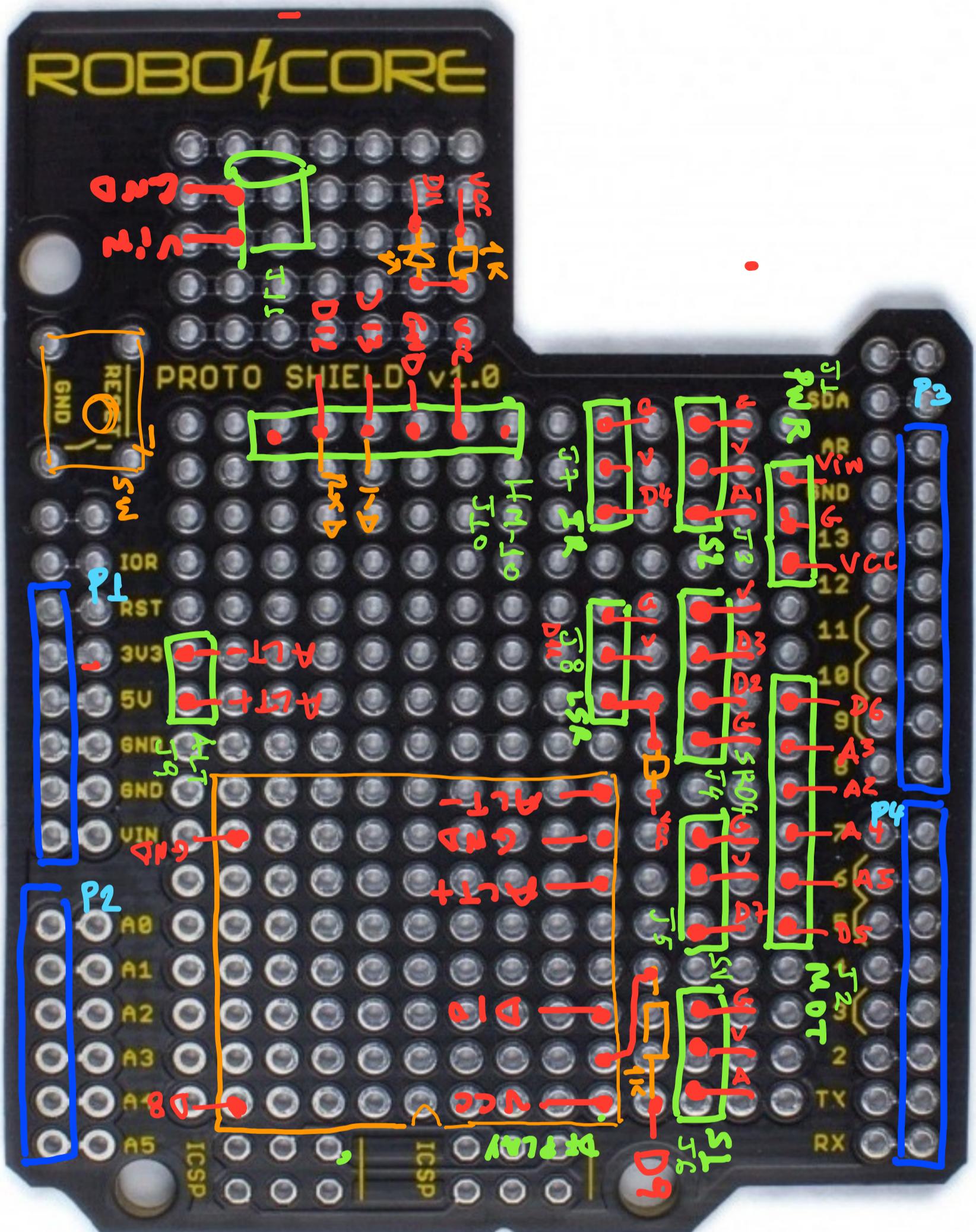


CARREGADOR  
USB

OBS - As pilhas só são retiradas do robô  
e inseridas no carregador







## Criar componentes e lib pr Proteus

- Criar nova library

- usar package

USER DEV. BAK

• IDA

• LIB

USER PKG. LIB

• IDX

Copiar → colar em  
nova pasta

- Renomear

DFPDEV. LIB

BAK

IDX

.

DFPPKG. LIB

• IDX

não pode estar  
somente leitura  
"propriedades"

- Copiar e recolocar na lib do proteus

- Adm Proteus

- Esquemático aberto

↳ library - library manager

↳ procurar a library criada - DFPDEV

↳ deletar os components que não quer

- Vai pr PCB Layout

↳ modo componente

↳ library - library manager

↳ procurar DFPPKG

↳ deletar os PKG que não quer

↳ procurar na library package e copiar

↳

# Criar Componente Proteus chave tctil

- 1- modelos 3D - site grabcad.com
  - ↳ Tactil switch
  - ↳ arq. STEP
  - ↳ copiar e colar na pasta MCAD
- abre proteus
- no esquematico
  - ↳ usar componentes SW-DIST-MOM
  - ↳ inserir chave no esquematico
- no PCB Layout
  - ↳ Seleciona Round-Through-hole Pad Mode - tam 100-45
  - ↳ posiciona de acordo com o datashet. do componente (olhar a numeracao dos pinos)
- Open 2D-Graphics mode definir o contorno do componente (desenhar a caixinha)
- Seleciona - botar direto - make package
  - ↳ Dar nome - CHAVÉ-TATIL-CURTA
  - ↳ Salvar na USERPKG
    - ↳ CATEGORIA - Miscellaneous
    - ↳ TIPO - Through hole
    - ↳ SUBCATEGORIA - Switches
  - ↳ Em 3D Mechanical Model
    - ↳ atribuir o modelo 3D baixado (.STEP)
    - ↳ MCAD → arq. STEP
    - ↳ posicionar corretamente
- Volta ao esquematico - editar componente
  - ↳ modo silencioso - botar direto - Decompose
  - ↳ edit properties - numerar os terminais

- Seleciona todo - Hacer diseño - make device
  - ↳ alterar o nome - Chave\_Tactil - curta, NEXT
  - ↳ atribuir o Package - ADD - Chave\_Tactil - curta
  - ↳ Fazer associação dos pinos - aceita PKG, NEXT
  - ↳ alterar o nome - Chave\_Tactil - curta, NEXT
  - ↳ salvar o DataSheet - USERNDEV
  - ↳ Categoría - Subcategoría - Fabricante
- ↳ Volver a PCB Layout
  - ↳ delet chave
  - ↳ Colocar line-to - Place-Component - SNT
  - ↳ posiciona → vai aparecer legende
    - ↳ o package estar associado ao componente

# PCB - Proteus Aras

## 1 - Escrever na placa

- ↳ Top Silk
- ↳ Seleciona  $\sqrt{A}$   $\rightarrow$  2D Graphics Text Mode
- ↳ escreve, depois arrasta

## 2 - Fazer Trilhas

- ↳ Seleciona   $\rightarrow$  Design Rule Manager (superior direito)
- ↳ Net Classes
  - ↳ Power e Signal
  - ↳ Power
    - ↳ Pair à Hz  $\rightarrow$  Both Cooper
    - Vari - Both cooper
  - ↳ Routing Styles (expressões dentro das trilhas) T40 ~ T200
  - ↳ Signal  $\rightarrow$  Both cooper
    - ↳ trilhas T40
  - ↳ Auto Rooter (sup)
    - ↳ Begin Router
    - ↳ Pode repetir a alteração.

## 3 - Output

- ↳ Gerber / Excellon Output
- ↳ faz verificação de erro

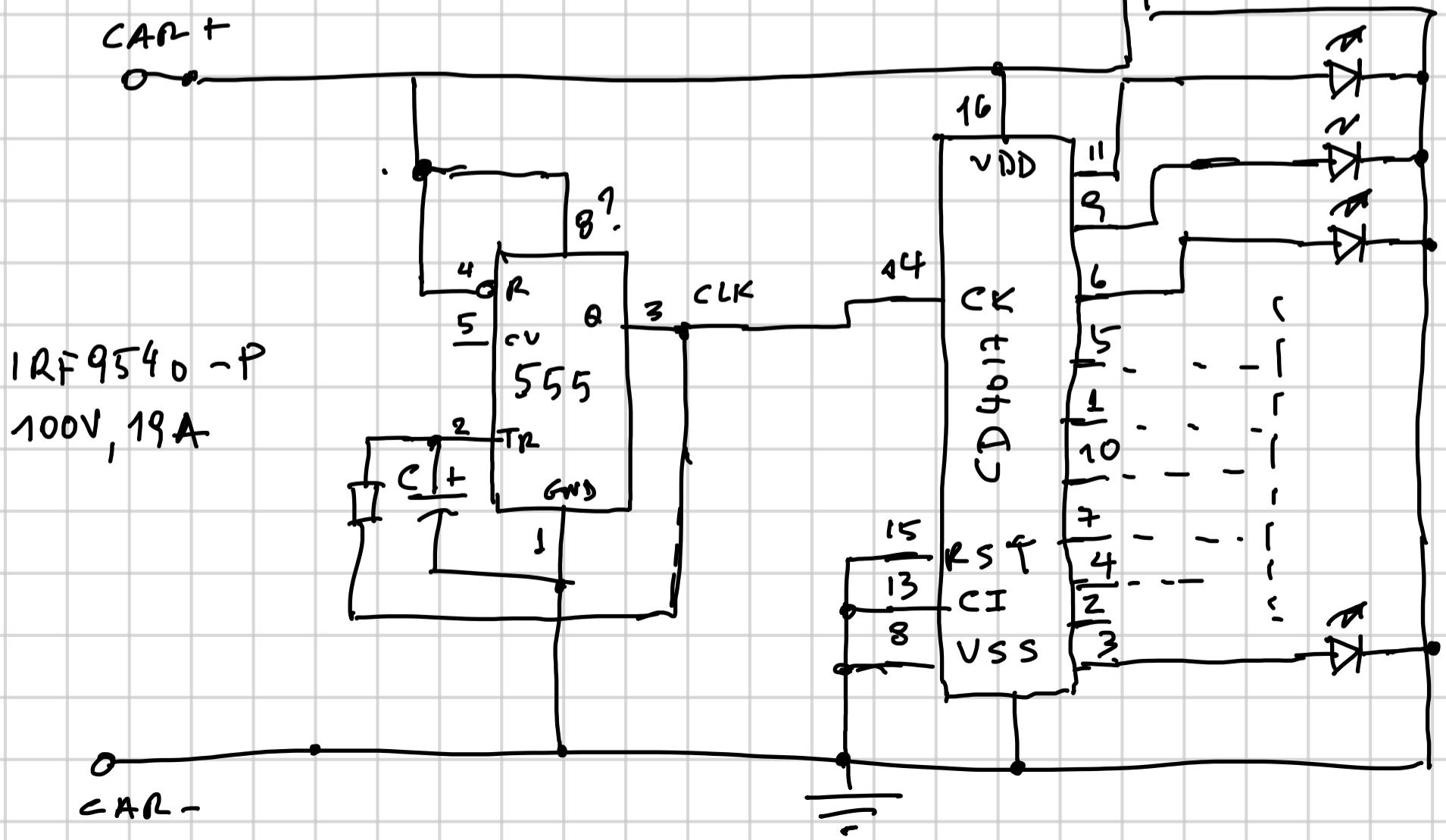
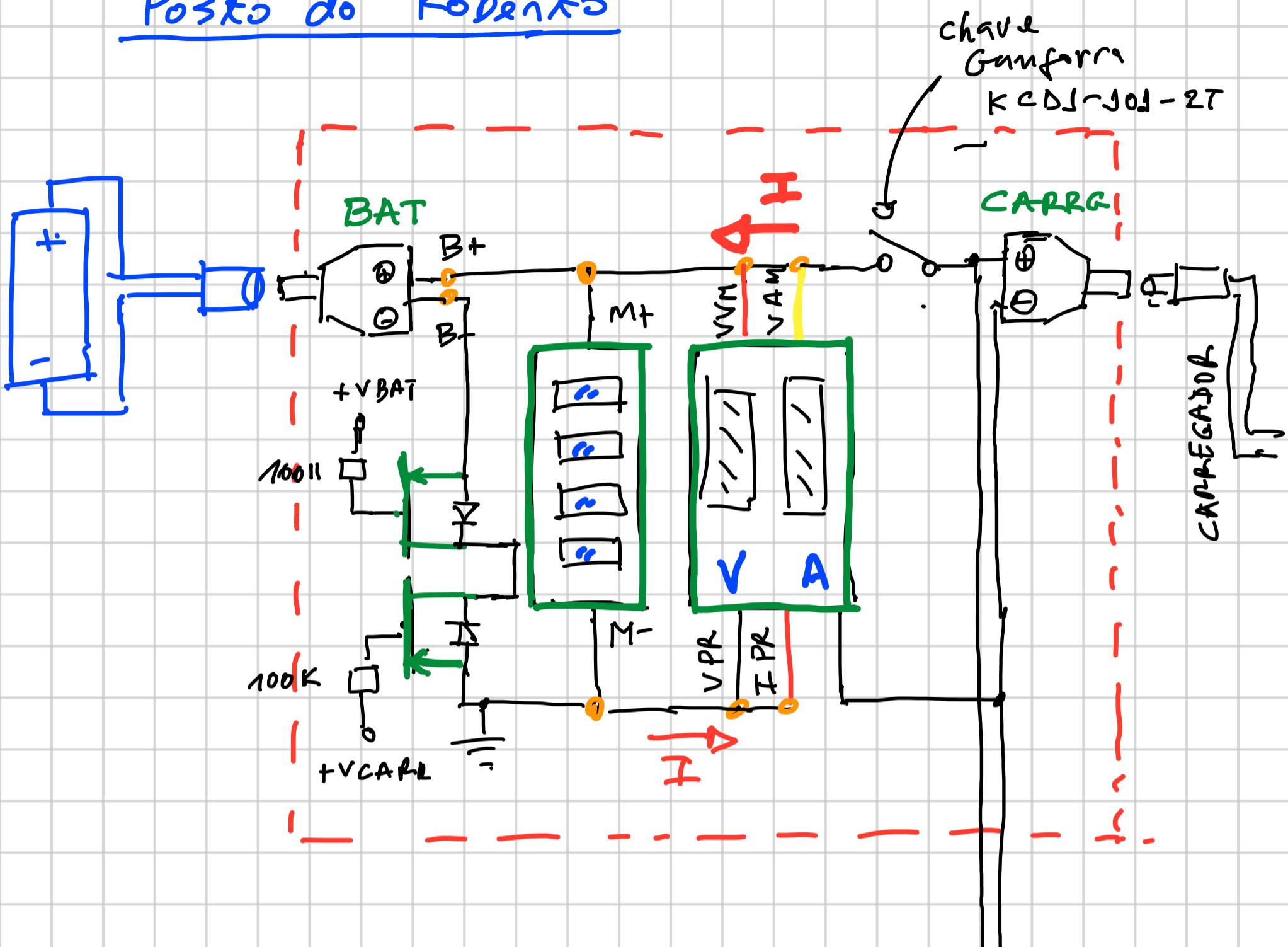
## 4 - Pedi-lo na PCBway.com

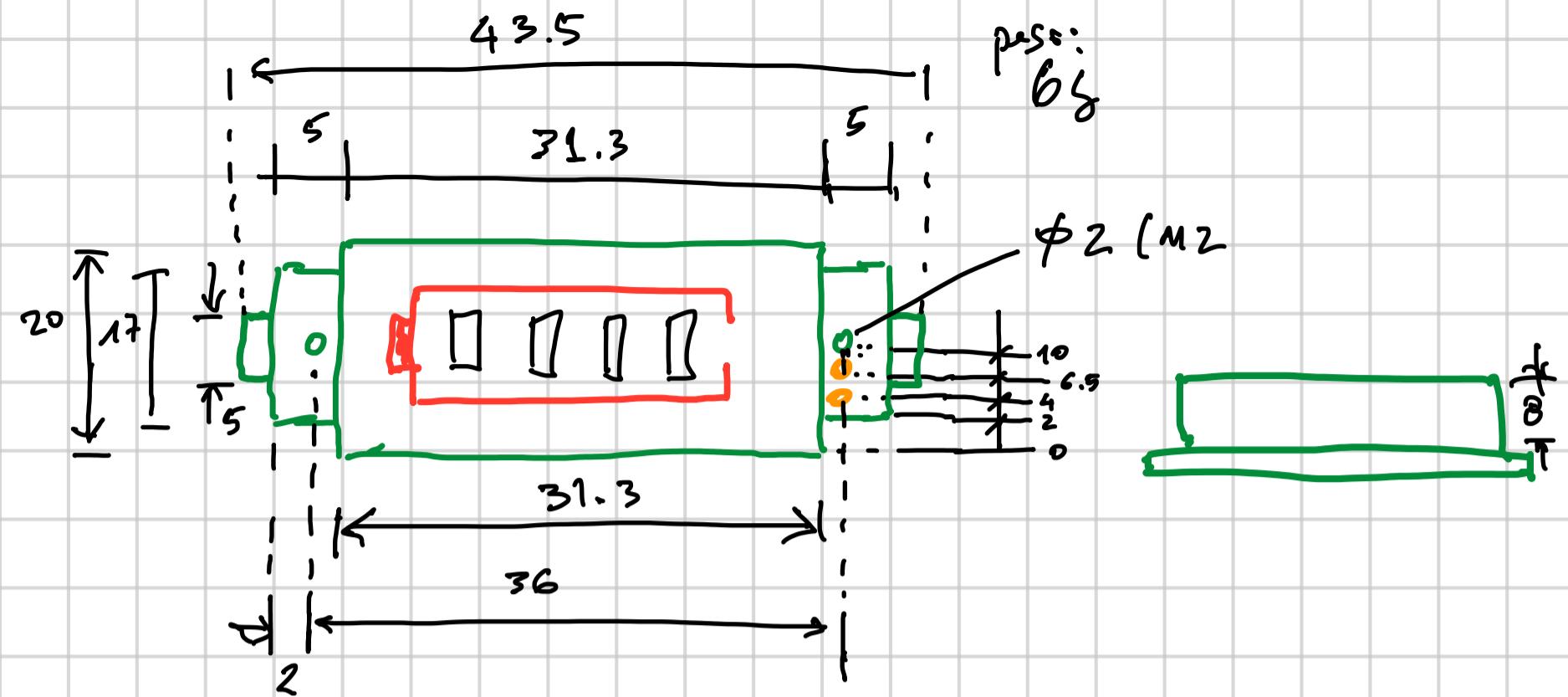
- ↳ adicionar arquivo Gerber

## 5 - Para descobrir as dimensões da placa

- ↳ Selecionar dimension mode  $\sqrt{\square}$
- ↳ google TH  $\rightarrow$  mm

# Posto do Roberto





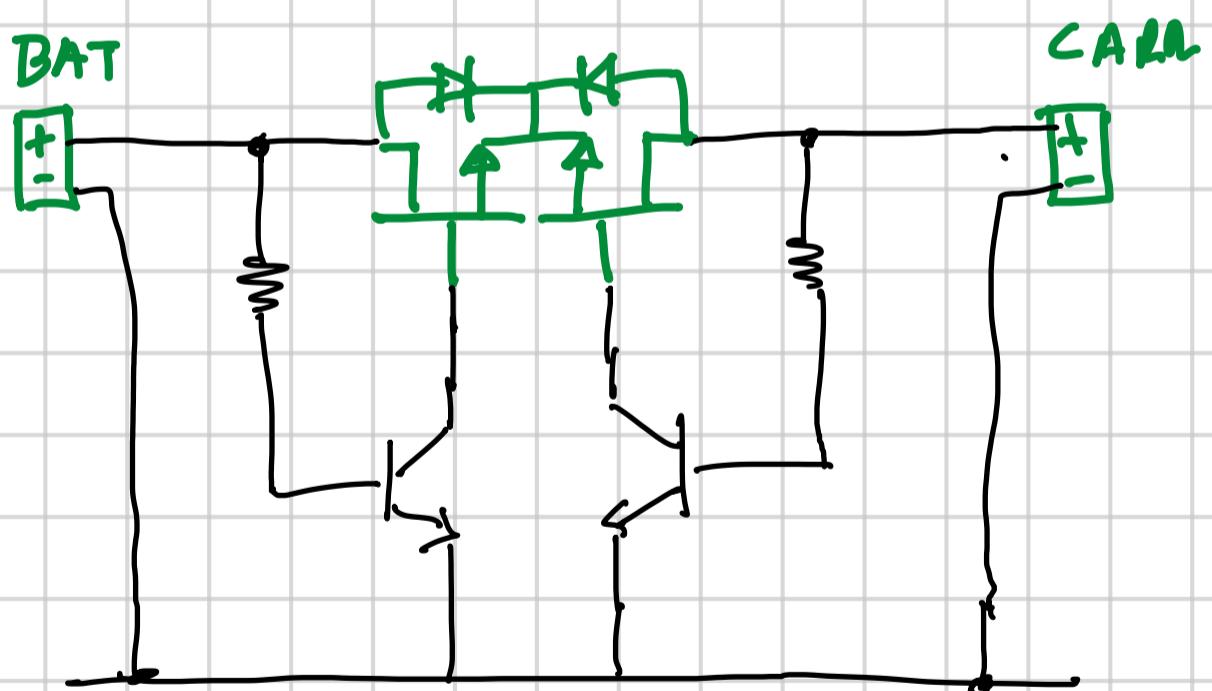
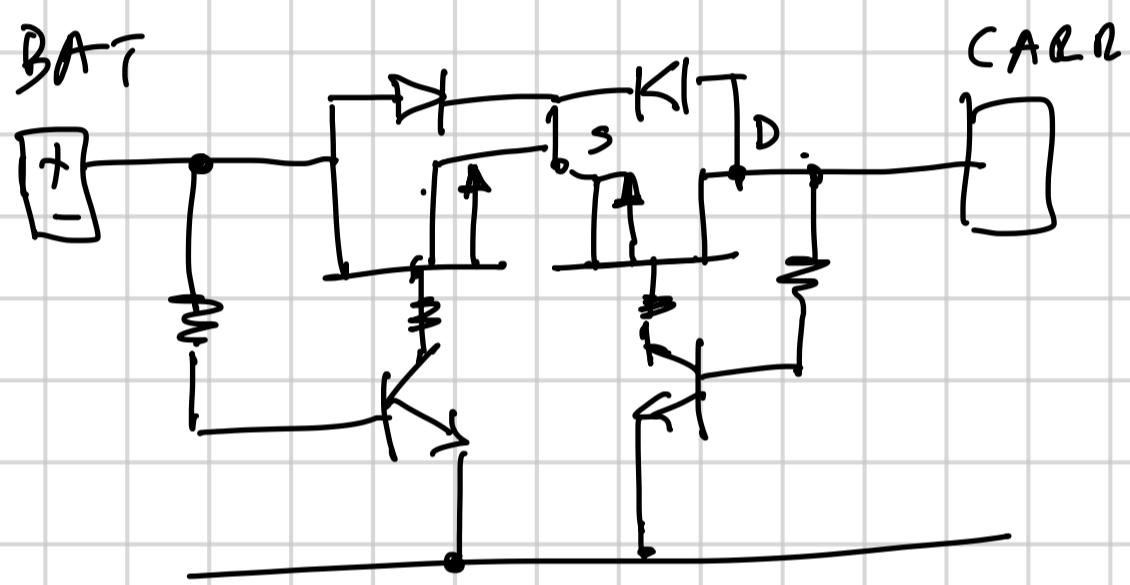
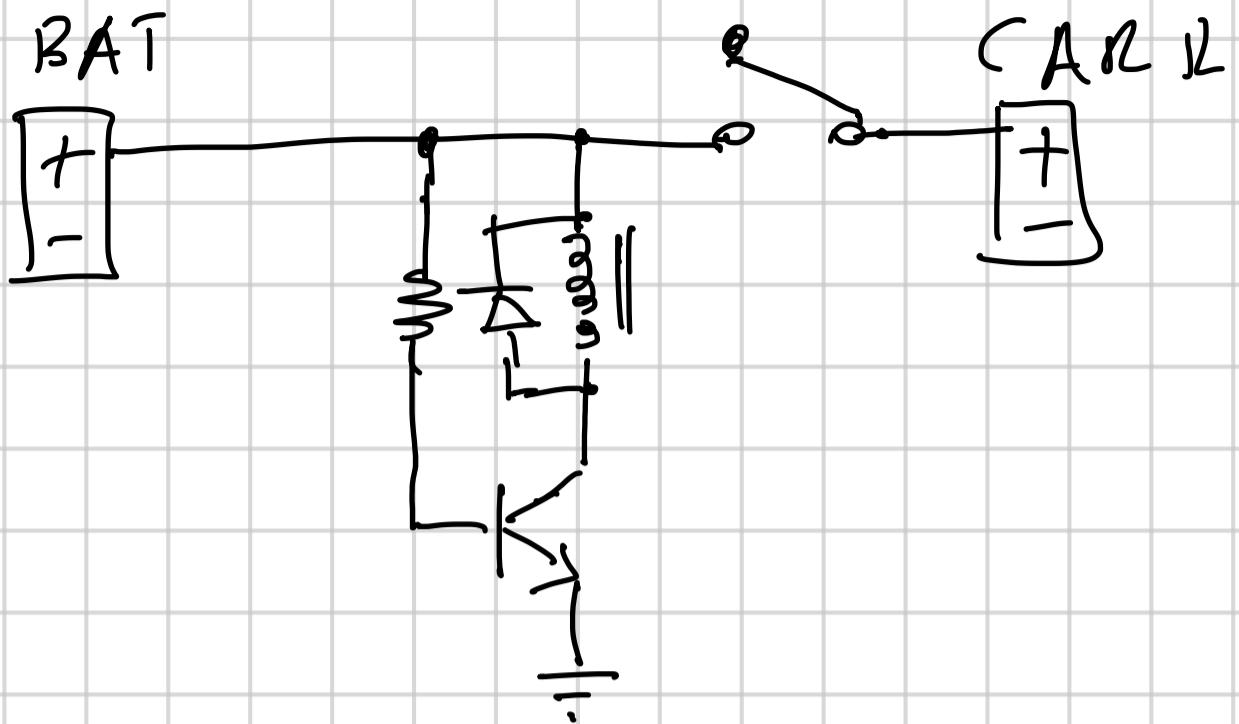
IRF840 - 8A 500V

IRF830 - 4.5A 500V

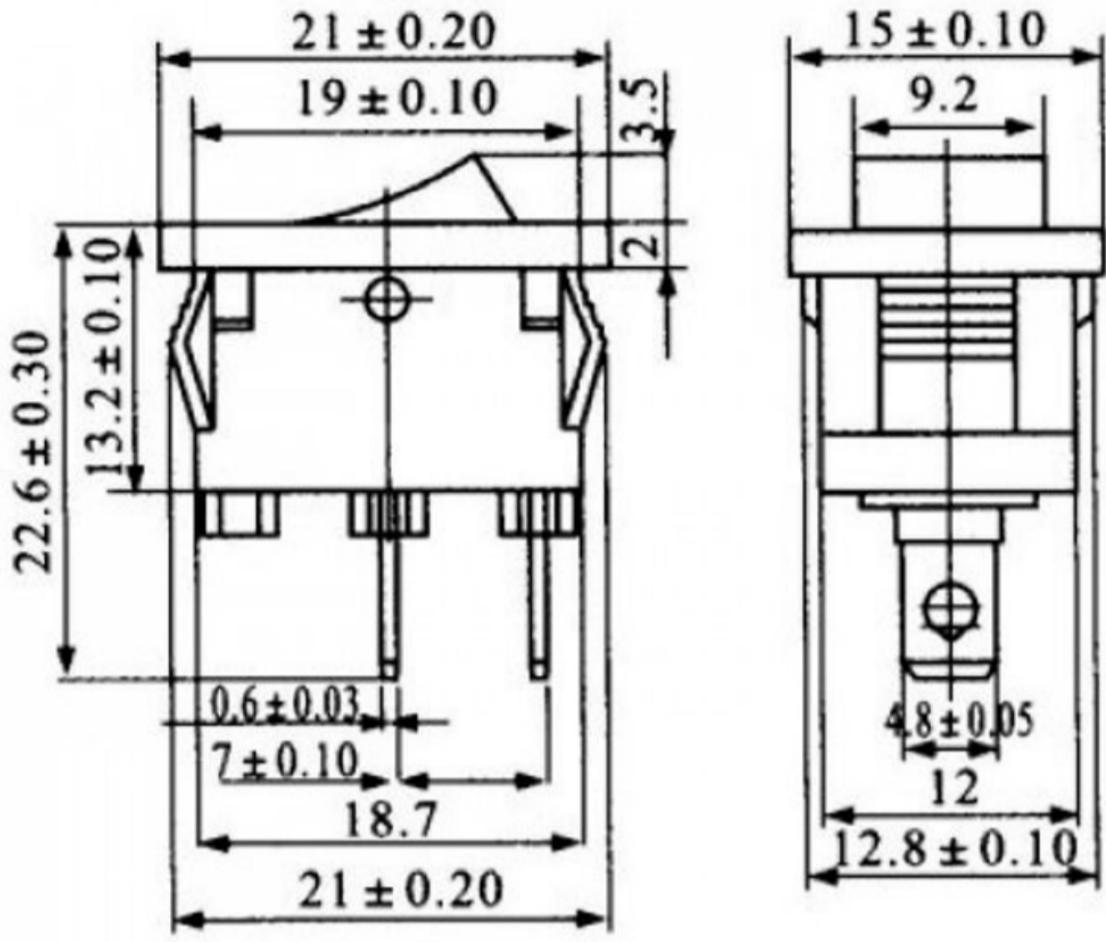
IRF540 - 28A 100V

IRF620 - 5.2A 200V

IRF630 - 9A 200V



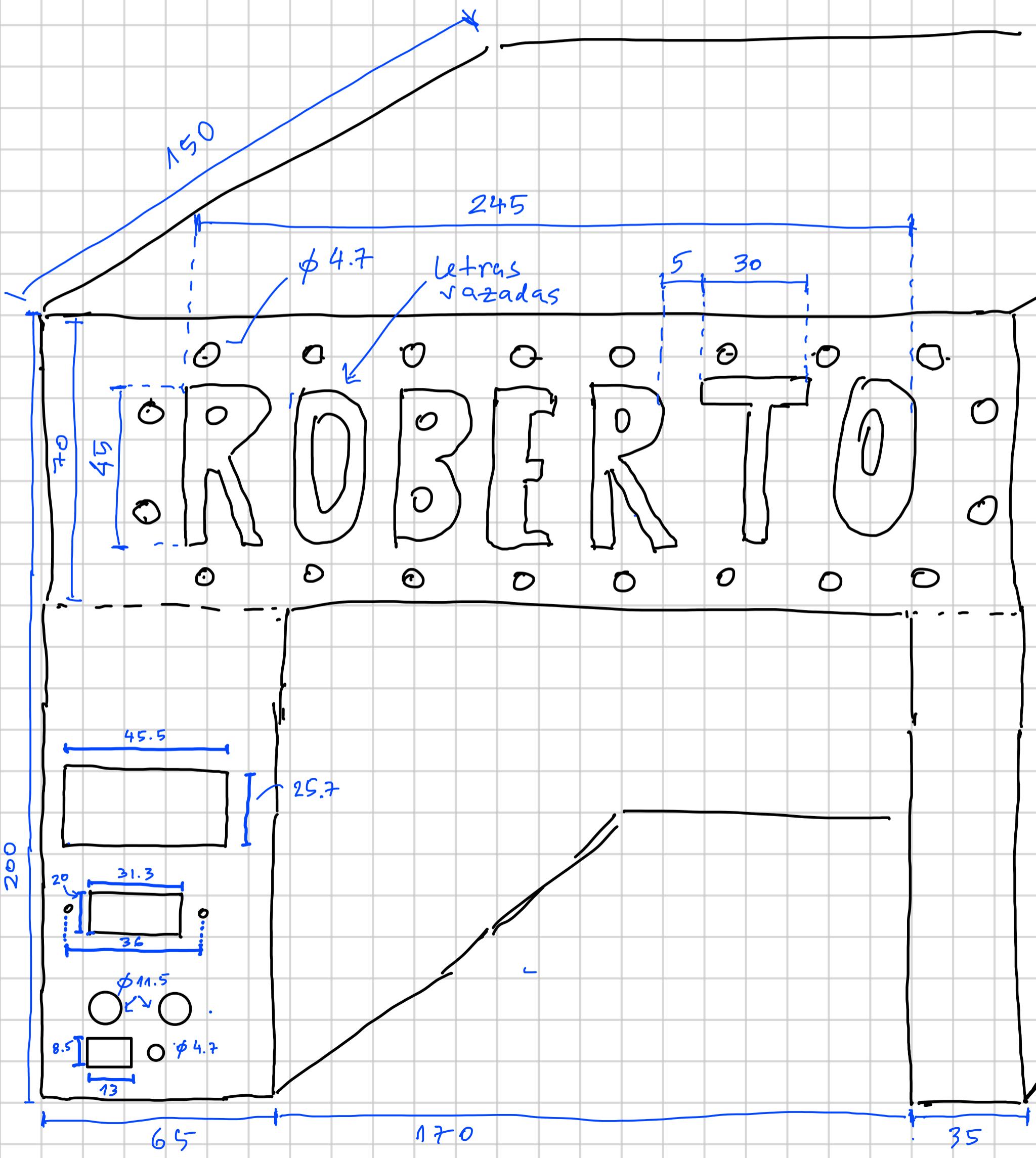
# Chave SPST - KCD1-101-2T



Posto do Roberto

Fabiana

98558 - 4434





13/12/20 gl

Paul McWhorter - Lesson 5: Prog. Core DC Motor Control Functions

- 1- Colocar as funções depois do void loop;
- 2- forward (float d)
  - ↳ d - unidades de distância
  - ↳ delay ( $d \times K \times 1000$ )  $\hookrightarrow$  deve ser calibrado para a distância desejada em 1 segundo
- 3- backward ( float d )
  - ↳ delay ( $d \times K \times 5000$ ) → idem.
- 4 - Criar funções para Calibração ou hardware
  - ↳ CalF, CalB, CalR, CalL
  - ↳ Aciona por 5s [ delay (5000) ] e para
    - ↳ melhor reduzir o delay para calibrar ângulo
  - ↳ mede a distância percorrida em 5s e calcula a velocidade de deslocamento
  - ↳ mede o ângulo percorrido e calcula a velocidade angular de giro
- ! 5- Precisa alimentar o motor com velocidade constante
  - ↳ vai precisar de um buck.

Lesson 6 :

- 1- Como fazer com que a distância percorrida não dependa da velocidade programada
  - ↳ identificar o  $VR_{min}$  (0-255) p/ funcionamento regular do motor  $\rightarrow$  Aprox. 500

- ↳ medir a distância percorrida em 5s para  $wv_{min}$  (100) e  $wv_{max}$  (255)
- ↳ a partir da distância calcular a velocidade para  $wv_{min}$  e  $wv_{max}$
- ↳ considerar a linearidade das funções entre  $wv$  e  $d$ .

↳ calcular  $m = \frac{\sqrt{255} - \sqrt{100}}{wv_{255} - wv_{100}}$

↳ identificas funções:

$$v - \sqrt{100} = m (wv - wv_{100})$$

↳ incluir o cálculo de  $v$  (a partir de  $wv$ ) no cálculo do tempo de percurso nas funções FWD e BKWD

### Lesson 8 :

1- Inverter a lógica e calcular  $wv$  necessário para atingir determinadas velocidades (entre  $v_{min}$  e  $v_{max}$ ).

↳ considerando a eq. do vídeo:  $v = 0.0075wv + 0.35$

tem-se:  $wv = \frac{(v - .35)}{.0075}$

2- No Sketch:

↳ definimos  $v$

↳ calculamos  $wv$

↳ Setamos a velocidade do motor c/  $wv$

↳ Calculamos  $t$  para distância  $d$

$$t = \left( \frac{d}{m} \right) * 1000$$

!

3- 2 fontes de erro na Calibragem  
1º linearidade 2º carga da bateria

↳ outras fontes de erro:

↳ diferença na aderência dos pneus

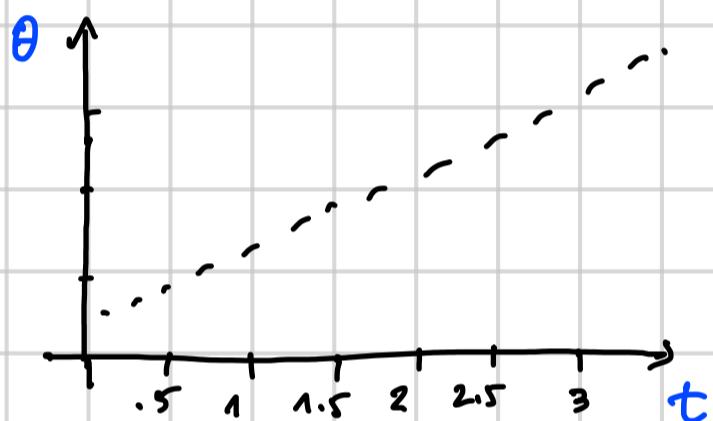
↳ pneus deformados

## Lesson 8: Calibrar o giro p/ direita e esquerda

1 → pode-se fazer por tentativa e erro,

↳ para algumas velocidades, ajustar o tempo p/ o giro de  $90^\circ$

2 → pode-se fazer por regressão linear



$$\theta = w * t + \theta_0$$

- é uma eq. linear da  $\theta$  em relação a  $t$ .  
com  $w$  fixo.

!

3 - ajustar calR e calL para receber wV

↳ girar a uma velocidade ( $wV$ ) menor, reduz  
a possibilidade de o pneu derrapar

! ↳ alterar as funções de Giro para reduzir a velocidade antes de girar, fazer o giro e estabelecer a velocidade normal.

! ↳ parar o carro, antes de fazer o giro.

4- Fazer as medidas do ângulo para os tempos de .5, 1, 1.5, 2.0, 2.5 e 3 segundos.

↳ no exemplo: 66, 135, 195, 270, 330, 405

↳ O resultado é approx. linear

5- Usar microsoft Excel p/ obter a equação

- 5- Usar microsoft Excel p/ obter a equação
- ↳ marcar tabela;
  - ↳ inserir gráfico de dispersão
  - ↳ clicar em um dos pontos (manuse direito)
  - ↳ inserir "linhas de tendência"
  - ↳ no painel, confirmar "Linear"
  - ↳ no painel, "Exibir Equações no gráfico"
  - ↳ Com esses dados a equação é:

$$y = 136,29x - 6$$

!

- ↳ desse modo, tem-se o  $t$  necessário para girar o carro um determinado  $\theta$

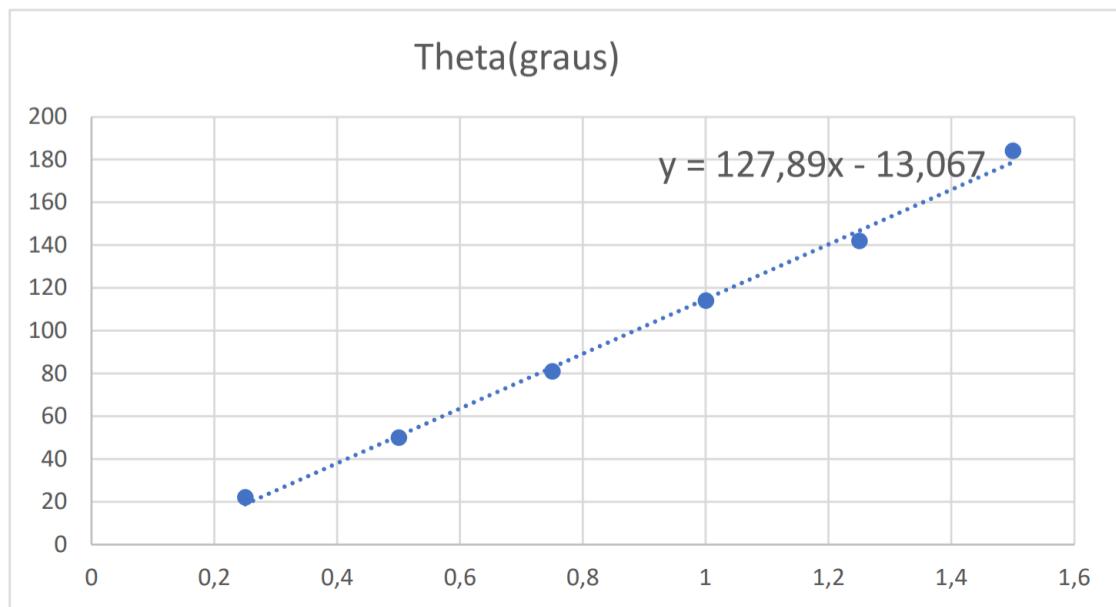
## 6- Modificações feitas c/ o Robert.

tempo(s)	Theta(graus)
0,25	22
0,5	50
0,75	81
1	114
1,25	142
1,5	184

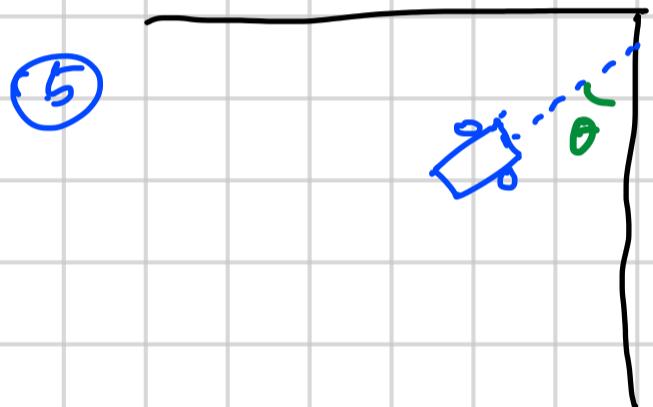
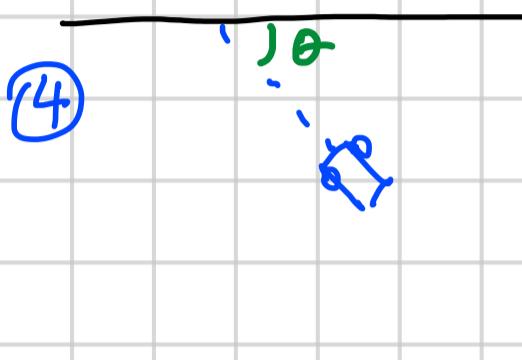
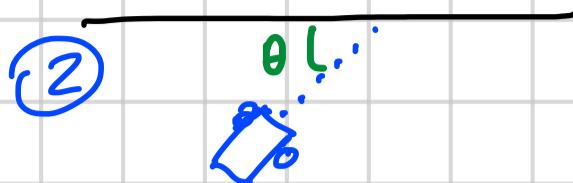
$$\theta = 127,89t - 13,067$$

ou

$$t = \frac{\theta - 13,067}{127,89}$$



## Situações da Decisão do Roberto



### ALGORITMO:

Situações

Decisão

2 e 4



giro de  $45^\circ$  para lado de  
maior distância

1 e 3



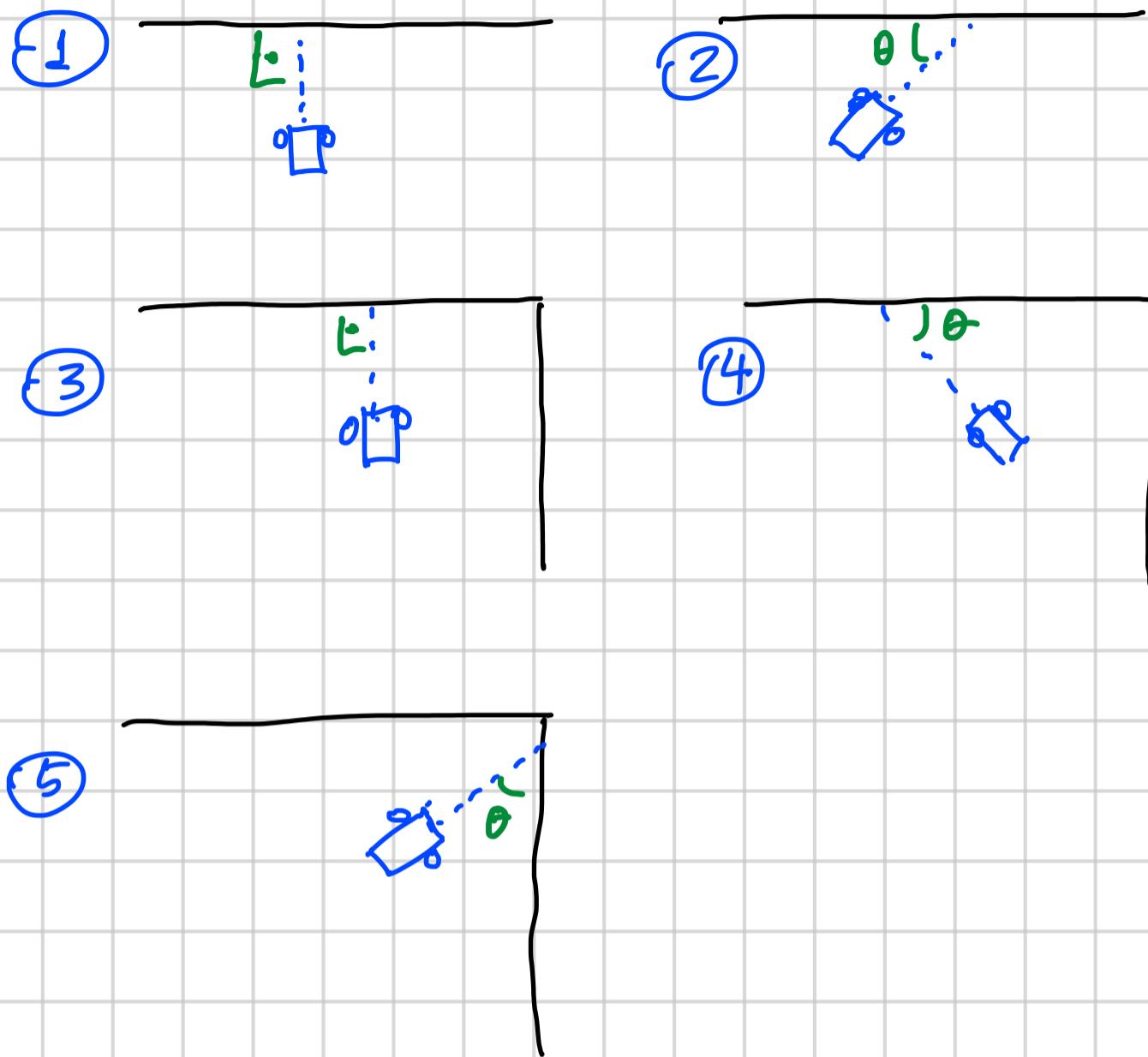
giro de  $90^\circ$  para lado de  
maior distância

5



marca né e giro  
de  $135^\circ$  se lado maior dist.

## Situações da Decisão do Robot.



SIT	Esq	Desq	Front	Ddir	Dir	Obs.
1	+	x	(-)	x	+	dist. Frontal < dist. laterais
2	-	(-)	-	+	+	pelo menos 1 dist. lateral
2	+	+	-	(-)	-	é >> que as outras dist.
3	-	-	(-)	+	+	pelo menos 1 dist. lat. >> que as outras. dFrontal é menor
	+	+	(-)	-	-	
4	-	(-)	-	+	+	igual a 2
	+	+	-	(-)	-	
5	-	(-)	-	-	-	todas dist. pequenas. Sendo a menor 1 diagonal.
	-	-	-	(-)	-	

Sit	Esq	Dsq	Front	Ddir	Dir	Obs.
1	+	x	-	x	+	dist. Frontal < dist. laterais
2	-	-	-	+	+	pelo menos 1 dist. lateral
2	+	+	-	-	-	só > que as outras dist.
3	-	-	-	+	+	pelo menos 1 dist. lat. > que as outras. dFrontal é menor
3	+	+	-	-	-	
4	-	-	-	+	+	igual a 2
4	+	+	-	-	-	
5	-	-	-	-	-	todas dist. pequenas. Sendo a menor 1 diagonal.
5	-	-	-	-	-	

### Algoritmo

→ grande  
 → pequena

1) Se Fronte = menor → Sit. 1 ou 3

Se Esq = e Dir = → Sit 1

Se ( Dir = e Esq = )

|| ( Dir = e Esq = ) → Sit 3

Se ( Dir = e Esq = ) → Sit. 5

2) Se ( Dsqf = || Ddir = ) → Sit = 2, 4, ou 5

Se ( Esqf = e Dir = ) → Sit. 5

else → Sit 2, 4

### Implementação

3 funções:

1º ler as 5 distâncias

2º identificar situações

3º executar o protocolo para a situação

Encontrar (máximo) valor de um vetor.  
ou  
main

int lista[5] = {3, 18, 2, 51, 45}

```
int main ( int - lista[5] ) {  
    int i;  
    int maior = lista[0];  
    for (i=1; i<5; i++) {  
        if (maior < lista[i]) maior = lista[i];  
    }  
    return maior;  
}  
  
int menor = lista[0];  
for (i=1; i<5; i++) {  
    if (menor > lista[i]) menor = lista[i];  
}  
idx_menor = i;  
return menor;  
idx_menor;
```