

# PROJECTE 3: IOT

Sergio Cortes Marquez

Jose Pineiro Yagües

Rubén Utrilla Fernández

# Índex

<b>Repartiment treball</b>	<b>2</b>
<b>Muntatge del cotxe</b>	<b>2</b>
<b>Raspberry PI</b>	<b>3</b>
<b>Connexió per ràdio dels Microbits</b>	<b>4</b>
<b>Node red i Telegram</b>	<b>5</b>
<b>Codi complet</b>	<b>8</b>
<b>Funcionament</b>	<b>10</b>

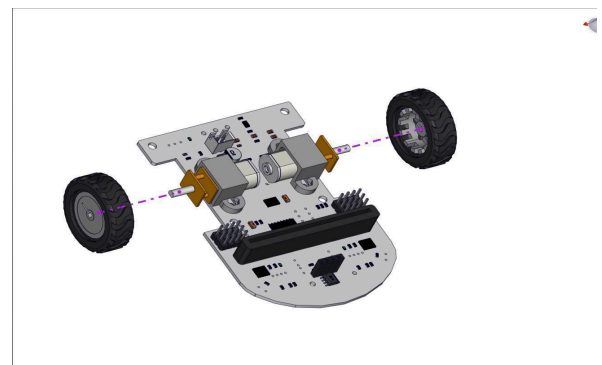
# Repartiment treball

- Jose
  - Encarregat del muntatge del cotxe. Això implica acoblar físicament totes les parts del cotxe, connectar la placa Micro:bit en el vehicle i assegurar-se que tot estigui funcionant correctament en termes de maquinari.
- Rubén
  - Responsable de la documentació. La seva funció ha estat documentar tot el procés, des del muntatge del cotxe fins a la configuració del bot de Telegram i la comunicació entre les plaques Micro:bit i la Raspberry Pi 4. Aquesta documentació pot incloure manuals de muntatge, guies de configuració, diagrames de connexió, etc.
- Sergio
  - Responsable de configurar el programari Node-XARXA en la Raspberry Pi 4, programar el bot de Telegram, establir la comunicació entre les plaques Micro:bit i la Raspberry Pi 4 a través de radiofreqüències, i assegurar-se que tot el sistema funcioni correctament.

Amb aquesta distribució de funcions, cada membre del grup ha contribuït de manera significativa al projecte.

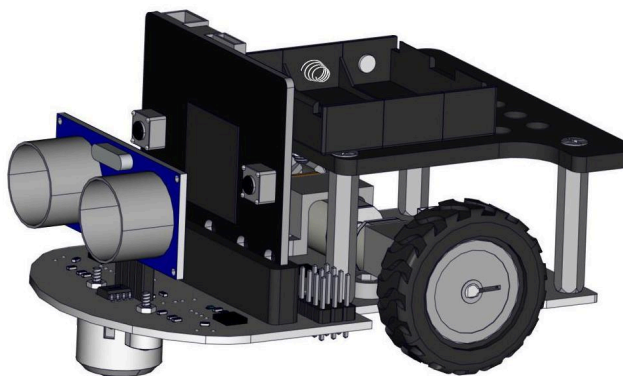
## Muntatge del cotxe

En primer lloc, vam procedir amb l'acoblament del vehicle, en el nostre cas, la part de la placa base de l'automòbil ja estava muntada. Per tant, el primer pas va consistir a instal·lar les dues rodes en cada costat.



Una vegada completat aquest pas, vam procedir a muntar la base del component de les piles.

Finalment, col·loquem les piles segons les indicacions proporcionades, juntament amb un cable que hem connectat a la placa base del vehicle.



## Raspberry Pi

El primer que hem fet per la instal·lació de la raspberry a sigut descarregar [el programa de raspberry](#) i [aquesta ISO](#), hem iniciat el programa anteriorment mencionat i hem configurat el wifi de l'institut i l'idioma del teclat, flashejem la targeta micro SD i ja tindrem llest el sistema operatiu dins de la raspberry pi 4.

Per instal·lar screenfetch entrarem a la terminal de la raspberry i farem:

```
sudo apt update  
sudo apt upgrade
```

Reiniciem amb:

```
sudo reboot
```

I instal·lem screenfetch amb:

```
sudo apt install screenfetch
```

Aquesta aplicació serveix per veure la informació del nostre sistema complet a partir del terminal

Per executar l'aplicació:

```
screenfetch
```

```
ies@raspberrypi:~$ screenfetch
.,,:cc;,'.      .,:cc;,'.
,ooocloooo: 'ooooocloo:
,loooc;;:ol :oc;;:oooo'
;ooooo: ,oooooc.
.,,:,'.      .,:,'.
.dQ. .dQoQo. 'oQ.
.oQo' 'oQoQo' 'oQo.
'' .odo. .odo. ''
. 'oQoQoQ' .oQoQoQ.
.oQ .oQoQoQoQ 'oQoQoQb. oQ.
:Qo QoQoQoQ 'oQoQo Qo'
'o 'oQo' .oQo. 'o' 'Q'
.oo. .oQoQo. .oo.
'oQoQo. 'oQoQoQo. .oQoQo
'oQoQo. 'oQoQo' .oQoQo
'oQo' .. 'oQ.'
. oQoQoQ.
'oQoQo'
ies@raspberrypi:~$

ies@raspberrypi
OS: Raspbian 12 bookworm
Kernel: aarch64 Linux 6.6.20+rpt-rpi-v8
Uptime: 3h 9m
Packages: 1527
Shell: bash 5.2.15
Resolution: No X Server
WM: Not Found
GTK Theme: PiXFat [GTK3]
Disk: 4.96 / 626 (8%)
CPU: ARM Cortex-A72 @ 4x 1.80GHz
GPU:
RAM: 480MiB / 7810MiB
```

## Connexió per ràdio dels Microbits

Per això necessitem 2 plaques micro:bit. Aquestes plaques per si mateixes tenen funció de radi, però per fer que es comuniquin entre elles hem de posar una mica de codi des de la pàgina web [Microsoft MakeCode](https://makecode.microbit.org/). El codi que hem de posar a les plaques MicroBit és aquest, per la primera placa (que va connectada a la Raspberry Pi 4):

```
en iniciar
  estableix grup 68 de ràdio
  redirigeix el port sèrie a l'USB
  assigna la velocitat de transmissió en bauds 115200 del port sèrie
```

I a la segona placa (que va connectada al cotxe):

```
en iniciar
  estableix grup 68 de ràdio
```

Això fa que les dues plaques es comuniquin pel mateix grup de radi.

# Node red i Telegram

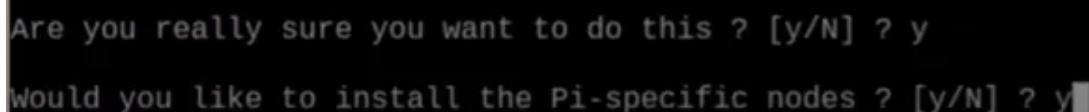
Per fer que el nostre cotxe funcioni a partir del nostre bot de Telegram hem hagut de fer diversos passos:

- **Descarregar Node-Red**

Per descarregar aquest software hem d'obrir una terminal a la nostra Raspberry i posar aquesta comanda:

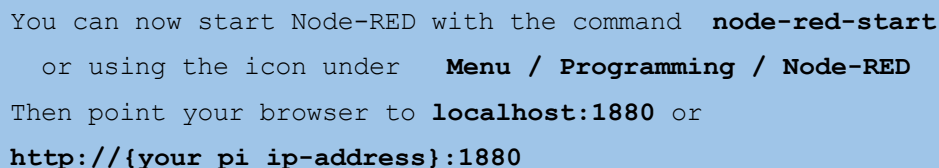
```
bash <(curl -sL  
https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update  
-nodejs-and-nodered)
```

Ens preguntarà si volem efectuar l'operació d'instal·lació i també la incorporació dels nodes específics i necessaris si volem interactuar amb els pins de la *Raspberry Pi*. Responem que **SÍ** a les dues preguntes.



```
Are you really sure you want to do this ? [y/N] ? y  
Would you like to install the Pi-specific nodes ? [y/N] ? y
```

En cas de finalització correcte, ens mostra la informació de com podem gestionar Node-RED.



```
You can now start Node-RED with the command node-red-start  
or using the icon under Menu / Programming / Node-RED  
Then point your browser to localhost:1880 or  
http://{your_pi_ip-address}:1880
```

Completem la configuració segons les respostes que es mostren a continuació:

```

=====
This tool will help you create a Node-RED settings file.

▼ Settings file · /home/docent/.node-red/settings.js

User Security
=====
▼ Do you want to setup user security? · No

Projects
=====
The Projects feature allows you to version control your flow using a local git repository.

▼ Do you want to enable the Projects feature? · No

Flow File settings
=====
▼ Enter a name for your flows file · flows.json
▼ Provide a passphrase to encrypt your credentials file · *****

Editor settings
=====
▼ Select a theme for the editor. To use any theme other than "default", you will need to install @node-red-contrib-themes/theme-collection in your Node-RED user directory. · default
▼ Select the text editor component to use in the Node-RED Editor · monaco (default)

Node settings
=====
▼ Allow Function nodes to load external modules? (functionExternalModules) · Yes

```

El settings file l'heu de crear a /home/el\_vostre\_usuari

Useu el terminal per engegar el servei:

```
node-red-start
```

Cal habilitar el servei de Node-RED per permetre que ens iniciï cada vegada que reiniciem la Raspberry. Executem al terminal:

```
sudo systemctl enable nodered.service
```

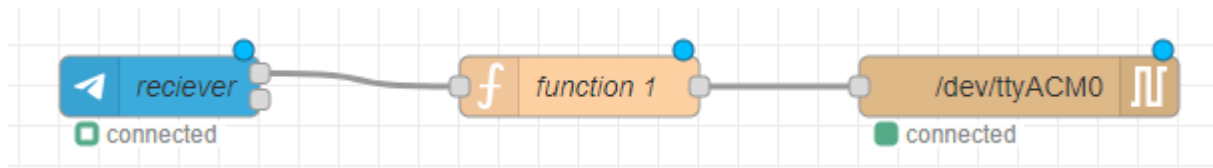
- **Crear el bot de Telegram**

Per crear un bot s'han de seguir les següents passes:

1. Comença un xat amb **@BotFather** a *Telegram*.
2. Envia l'ordre **/newbot** per crear un nou bot.
3. Segueix les instruccions de **@BotFather** per configurar el **nom**.
4. I el **nom d'usuari** del teu bot, posar el mateix que el nom però acabat amb **\_bot**.
5. Una vegada creat el bot, **@BotFather** et proporcionarà un token d'accés únic (*HTTP API Token*) que hauràs de desar per utilitzar-lo més endavant.

- **Crear el “flow” al software Node-Red**

El “flow” o etiquetes que hem de posar per configurar el nostre bot amb el Node-Red és aquest:



Dins del reciever de Telegram la configuració és aquesta:

The screenshot shows the 'Properties' panel for the Telegram receiver node. It includes the following fields:

- Bot:** A dropdown menu showing '@akinfenwa\_bot'.
- Name:** A text input field containing 'reciever'.
- Download Directory:** A text input field containing 'Download directory'.
- Filter:** A checkbox labeled 'commands (from configured command nodes)' which is currently unchecked.

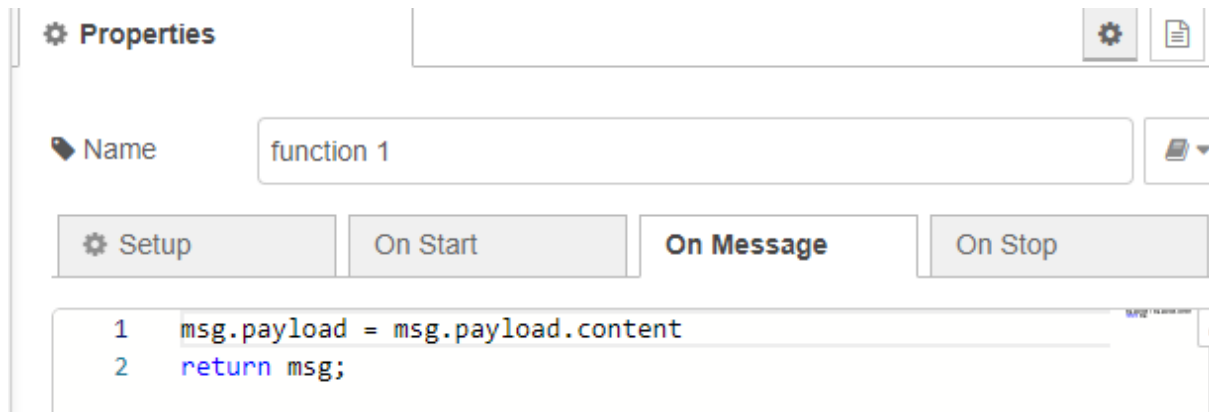
On la configuració del bot és aquesta:

The screenshot shows the configuration panel for the Telegram bot. It includes the following fields and sections:

- Bot-Name:** A text input field containing '@akinfenwa\_bot'.
- Token:** A text input field containing '7005743521:AAGGWesjAl6lBMRvmO5t6ACq8IsA89eGfQ'.
- Tip:** A yellow box with the text: 'Tip: If you don't have a token yet, you can create a new one here: @BotFather.'
- Users:** A text input field with the placeholder '(Optional list of authorized user names e.g.: hugo,sepp,egon)'.
- ChatIds:** A text input field with the placeholder '(Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)'.
- Server URL:** A text input field with the placeholder '(Optional URL for proxying and testing e.g.: https://api.telegram.org)'.
- Update Mode:** A dropdown menu showing 'Polling'.
- Polling Options:** A section containing a 'Poll Interval' input field with the value '300'.
- Tip:** A yellow box with the text: 'Tip: Polling mode is very robust and easy to set up. Nevertheless it creates more traffic on the network over time.'



L'etiqueta de funció té aquest codi on ens especifica que solament detecta el text que rep el bot.



I en l'etiqueta de “serial out” seleccionem l'única opció que ens sortirà, el que ens surti serà la Micro:Bit que tinguem connectada a la Raspberry Pi 4:



## Codi complet

- **Micro:Bit connectat a la Raspberry**

Hem posat el per sempre per fer que no funcioni solament una vegada, l'etiqueta de llegir la cadena del port sèrie fa que quan s'envia un missatge al bot de Telegram, el missatge s'assigna a la variable que hem creat anomenada “mando”. Després d'això posem les etiquetes de “si”, per fer la simulació de “if i else”. Això farà que depèn el missatge que s'envia al bot es transmetia un número diferent a través de ràdio al Micro:Bit del cotxe, on depenent el número que es transmet farà un moviment o un altre el cotxe.



- **Micro:Bit connectat al cotxe**

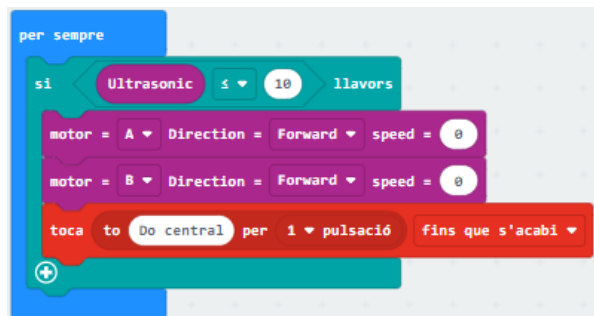
En aquest codi la seva funció és comprovar que li arriba mitjançant la ràdio, és a dir, depenent el valor que li arriba, el cotxe farà un moviment o un altre.

Al nostre cas les nostres “normes” són aquestes:

- 1 = El cotxe avança
- 2 = El cotxe s'atura
- 3 = S'encenen les llums
- 4 = S'apaguen les llums
- 5 = Gira a la dreta
- 6 = Gira a l'esquerra
- 7 = Va marxa enrere



Aquesta part de codi s'aplica sempre, i la seva funció és que funcioni el sensor d'aproximació, al nostre cas quan el sensor detecta algun objecte a distància igual o menor a 10, aquest s'atura i fa na senyal sonora, un xiulet.



## Funcionament

Hem aconseguit que el cotxe funcioni cap endavant, cap enrere, a la dreta, a l'esquerra, apagar les llums, encendre les llums i per últim hem configurat el sensor, a continuació deixaré un vídeo del seu funcionament:

