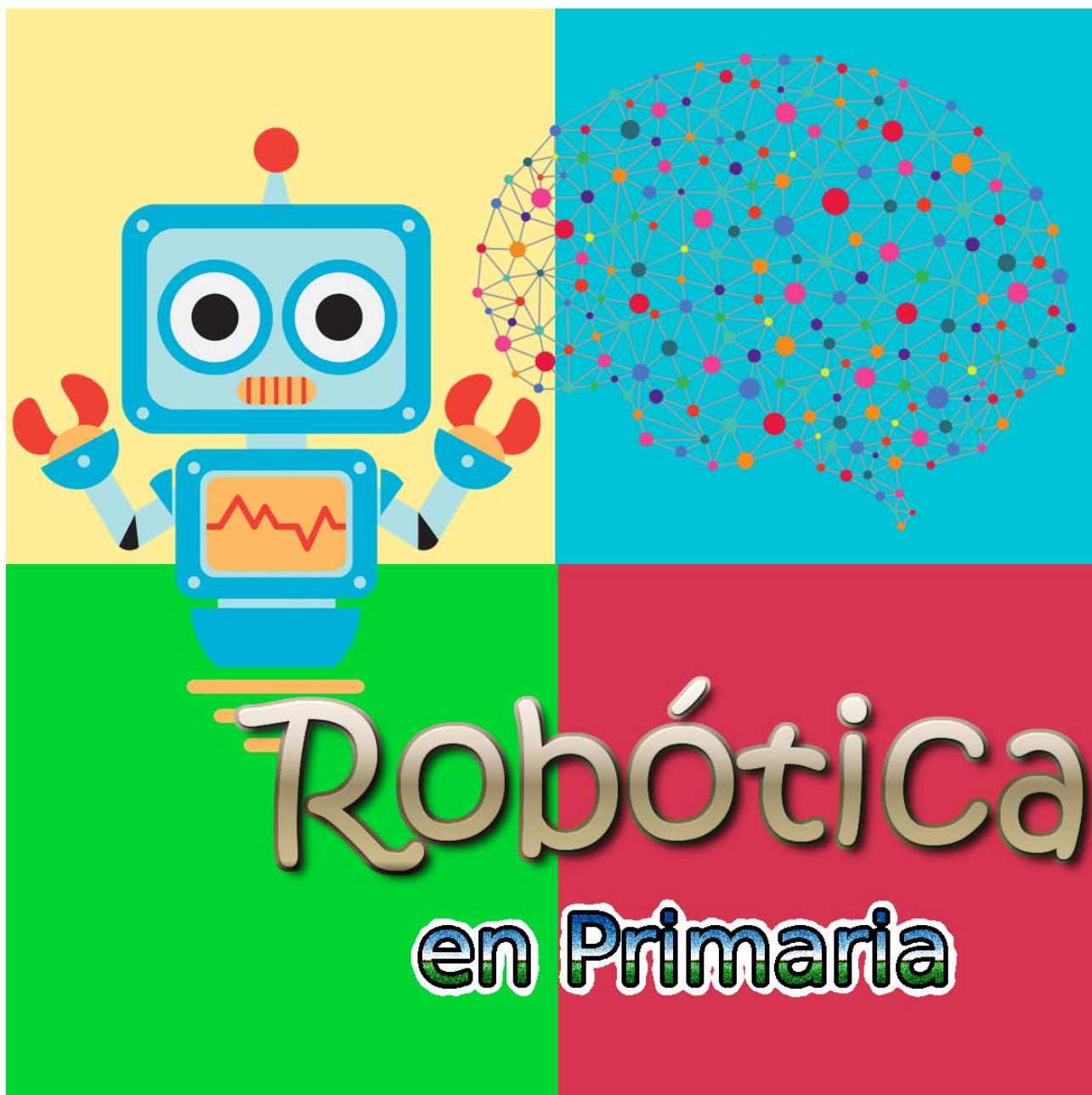




CFR-Ferrol- 2017-18

Reyes Montero Vale

Diferentes xeitos de programar o noso mBot

- 1) [mBlockly para mBot](#) (+4anos) nivel Básico. Entorno de desenvolvemento sinxelo para aprender o básico da programación con mBot



- a) é unha APP so dispoñible na APP Store para dispositivos IOS con versión 7.1 ou posterior. Compatible con Ipad
- b) Non está dispoñible en Español
- c) Control mediante Bluetooth

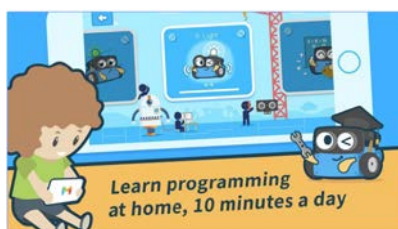


- 2) [mBlock](#) É un software de programación gráfico, deseñado por Makeblock para a educación en competencias STEAM.



- a) Dispoñible para dispositivos Android (tablets e teléfonos móbiles) e tamén para IOS (Ipad e Iphone).
- b) Control mediante Bluetooth.

E MOI RECOMENDABLE REALIZAR A HISTORIA, PARA FAMILIARIZARSE CO ENTORNO DE PROGRAMACIÓN



- 3) [Makeblock](#)



- a) Software de control de robots con dispositivos intelixentes. Permite usar o teléfono móbil ou tableta coma un mando a distancia (modo Xogo).
- b) Permite crear paneis de control remoto personalizados (modo Crear).
- c) rae as instrucións para construír o mBot e as distintas ampliacións.
- d) Permite controlar tanto o mBot, coma o mBot Ranger, Drons (AirBlock) e brazos robóticos (Starter e Ultimate)
- e) APP dispoñible en Android (tabletas e móbiles) e IOS (Ipad e Iphone)
- f) Control mediante Bluetooth
- g) Versión en Español

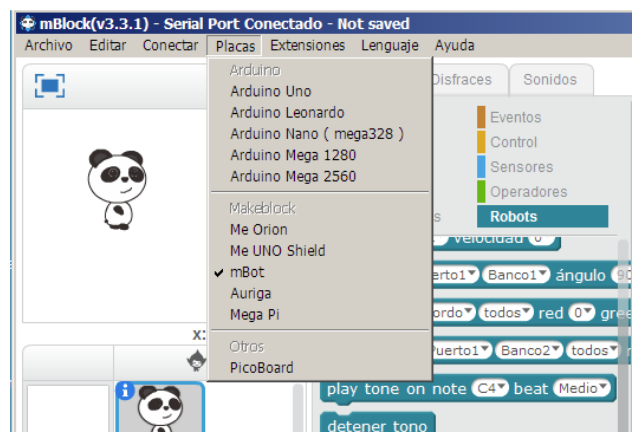


4) [Software mBlock para PC](#)

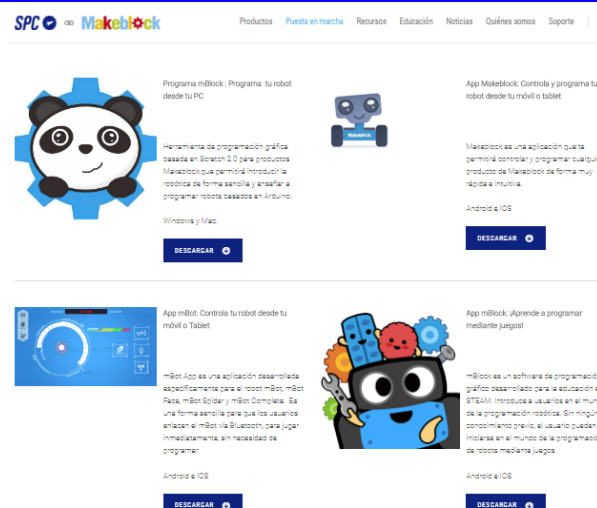
Entorno gráfico de programación baseado en Scratch 2.0 +Arduino. Interface (aparencia) moi amigable e intuitiva.



- a) Emprega bloques (de programación) previamente definidos para dar ordenes ao robot.
- b) Podemos descargar os programas de xeito inalámbrico do ordenador ao robot mediante (bluetooth) ou 2.4G.
- c) Para usuarios máis avanzados permite traducir os bloques de programación (Scratch) a código fonte de Arduino.
- d) Permite comprobar o programa en tempo real no robot ou ben descárgalo de xeito permanente.
- e) Sirve para todos os modelos (mBot, mBot Ranger, Ultimate Kit)
- f) Dispoñible par PC e Mac, e tamén en Linux. Pendente a instalación efectiva nos equipamentos Abalar (Edixgal)



[Dende aquí terás acceso as descargas de todas estas aplicación](#)



CONEXIÓN INALÁMBRICA

Por Bluetooth: Este robot incorpora un módulo Bluetooth 2.0 e 4.0, deseñado para a placa mCore. Compatible con smattphones Android e IOS, e tamén con ordenadores (que dispoñan de conexión Bluetooth)

Por módulo 2.4G (non incluído neste pack) dispón dun USB-serie inalámbrico para conectar aos PC, e outro módulo 2.4G para a placa mCore do mBot. Emprega a mesma tecnoloxía que un rato inalámbrico, moi axeitado para a aula, xa que non necesita drivers nin sincronización.

Enlaces moi útis

- 1) [Introdución á Robótica Educativa en Primaria](#) (**Aula Virtual do CAFI**- Centro Autonómico de Formación Integrada) Si xa sodes usuarios do CAFI; accedes pola marxe superior dereita da páxina e inclúides os vosos datos e a chave de automatriculación (**robotica17**) e xa estades preparado para descubrir todo o que teñen recollido.
- 2) Curso “[Proxectos de Robótica de Primaria- mBot](#)” da Aula Virtual do CFR de Ferrol
- 3) [Páxina oficial do Makeblock](#) , dende onde podedes atopar o software mBlock para descargar, acceso á tenda, foro, formación...
- 4) [Retos robótica; taboleiro](#) con diferentes enlaces a moitos retos para programar o mBot

Retos robótica



- 5) [Cableando o teu robot mBot](#); explicación visual e intuitiva de como conectar os diferentes compoñentes do robot á nosa placa controladora (portos, motores, adaptadores, servos, microswitches, bandas de LEDs, sensores temperatura, fontes de alimentación...)
- 6) [Programar o noso robot con Scratch+mBlock](#) Axuda a instalar os drivers para Arduino e Mac OSX, explica como conectar o robot ao ordenador mediante cable USB-porto Serie COM, conéctalo mediante Bluetooth. Seleccionar o tipo de placa controladora (mBot, Auriga..) Actualizar o firmware. Escribir e descargar os programas no robot. Tamén trae algúns exemplos sinxelos.
- 7) [Manual Kids-Maker](#)
- 8) [Manual de 145 páxinas “Divirtiendome con mBot” de Susana Oubiña](#)
- 9) [Manual de 37 páxinas de Javier Fernández Panadero](#)

ALGORITMO

Algoritmo: É un conxunto ordenado de **operacións, instrucións, ben ordenadas e finitas**, para realizar unha determinada actividade ou resolver un problema, sen que xere dúbidas. En programación defínese coma unha **secuencia de pasos lóxicos para solucionar un problema**.

É moi importante aprender a facer algoritmos antes de poñernos a programar; imos facer algún exemplo;

Exercicio 1.- Describe o máis detalladamente posible, e en orde, os pasos que debemos realizar para “fritir un ovo con puntillas”

1. Coller unha tixola e poñela sobre a cociña
2. Acender a cociña
3. Botar aceite ata que cubra o fondo da tixola e levante a lo menos 5 mm
4. Esperar a que o aceite estea suficientemente quente
5. Comprobar a temperatura con unha miga de pan
6. Abrir o ovo sobre o aceite quente
7. Engadirlle sal ao gusto
8. Cunha espumadeira botar aceite quente por riba do ovo
9. Se nos gusta para untar, non deixar fritir moito tempo
10. Coller un prato
11. Retirar o ovo coa espumadeira e servilo no prato
12. Apagar a cociña

Exercicio 2.- Constrúe un avión de papel paso a paso e de forma ordenada, a continuación escribe en orde os pasos, para que outra persoa que non sabe poida construílo.

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.

DIAGRAMA DE FLUXO

Un **diagrama de fluxo**, é a representación gráfica dun Algoritmo.

A diferenza dos seres humanos, que realizamos tarefas sen pararnos a pensar nos pasos que debemos seguir, as computadoras (robots) son moi ordenados, e necesitan que alguén os programe , e lles indique cada un dos pasos que deben facer e a orde lóxica de executalos.

Símbolos dos Diagramas de Fluxo



	Inicio / Final Emprégase para indicar o inicio ou final dun programa. So pode saír unha liña e ao final só pode chegar unha liña		Decisión Indica a comparación dos datos, e dependendo do resultado (Verdadeiro ou Falso) tómake un camiño ou outro no diagrama
	Entrada ou saída de datos. De momento só usaremos como entrada		Acción ou proceso Indica unha acción ou instrución que debe realizar o computador ou robot (operación, cambios de valores..)
	Saída impresa Presenta un ou varios resultados de forma impresa		Fluxo Indica a orde lóxica do diagrama, ou o sentido de execución das operacións.
	Saída en pantalla. Instrución para presentar mensaxes ou resultados na pantalla.		Entrada por teclado Instrución de entrada de datos por teclado. Indica que o computador ou robot debe esperar a que o usuario teclee un dato que se gardará nunha variable ou constante.

REGRAS BÁSICAS PARA ELABORAR DIAGRAMAS DE FLUXO

- Poñer un encabezamento que inclúa un título, a función do algoritmo, o nome do autor e a data de elaboración.
- Só se poden empregar os símbolos anteriores
- Os diagramas débense debuxar de arriba cara abaixo, e de esquerda a dereita.
- A execución do programa sempre comeza na parte superior do diagrama.
- Os símbolos de INICIO e FIN , deben aparecer unha soa vez.
- A dirección do fluxo representase por medio de frechas (liñas de fluxo)
- Todas as liñas de fluxo deben chegar a un símbolo ou a outra liña.
- Unha liña de fluxo nunca debe cruzar outra. Cando se crucen , unha debe estar arqueada.

9. Cada rombo de decisión debe ter como mínimo dúas liñas de saída (unha para o SI e outra para o NON)
10. As accións e decisións débense escribir empregando o menor número de palabras posible, para que non resulten confusas.
11. Todo diagrama debe ser claro, ordenado e fácil de recorrer.

É un xeito de representar instrucións por medio de palabras en forma de frases.

Exemplo: Seudocódigo para calculara a área dun triángulo rectángulo de base 3 cm , altura 4 cm e hipotenusa 5 cm;

1. Inicio
1. Asignar o número 2 á constante “división”
2. Asignar o número 3 á constante “base”
3. Asignar o número 4 á constante “altura”
4. Gardar na “variable área” o resultado de base x altura / división
5. Imprimir na pantalla o valor da variable “área”
6. Fin

<https://sites.google.com/site/aprendoscratch/s02>

MÓDULOS QUE PODEMOS CONECTAR Á PLACA

Os módulos que podemos conectar á placa veñen identificados e clasificados por **unha cor ID**. Esta cor **corresponde coa cor do porto RJ25** ao que debemos conectalo. Estes son algúns dos sensores dispoñibles.

SIGUE LIÑAS: Este módulo fai posible que o noso robot siga unha liña negra sobre o fondo branco e viceversa.

COR AZUL Interface **DIXITAL DUAL**, por defecto o **PORTO 2**



MATRIZ DE LEDs:

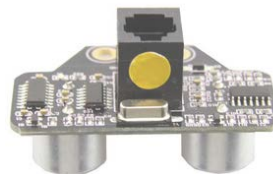
COR AZUL Interface **DIXITAL DUAL**, por defecto o **PORTO 4**

Olo coa montaxe, non facelo invertido!!



MÓDULO ULTRASÓNS: permite medir distancias dun máximo de 4 m e un mínimo de 3cm.

COR AMARELA; interface **DIXITAL**, por defecto **PORTO 3**



MÓDULO DISPLAY 7 SEGMENTOS: pódense visualizar 4 díxitos e o punto decimal

COR AZUL Interface **DIXITAL DUAL**, por defecto o **PORTO 4**



SENSOR PIR (de movemento) para detectar persoas ou animais nun rango de ata 6 m e nun ángulo de 120º. Se algo se move dentro deste rango, o sensor activa a saída dixital SIG en HIGH.

COR AZUL Interface **DIXITAL DUAL**



SENSOR DE GAS, capaz de detectar gas natural, propano, metano, alcol, hidróxeno, fume...(non incluído no KIT)

GRIS: porto Serie Bluetooth.



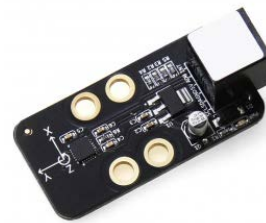
O módulo de **JOYSTIC** é unha palanca de control de mando, podemos empregalo para mover o un obxecto no escenario.

NEGRO: interface analóxica e dual.



ACELERÓMETRO e XIROSENSOR, pode medir o movemento angular e a aceleración nos 3 eixes, para recibir información dende o robot. (non incluído no KIT)

BLANCO porto I2C



POTENCIOMETRO (Resistencia variable) de 50KiloOhmios, permite converter un sinal analóxico de rotación (variación da resistencia) nun xiro do servomotor, variar o brillo dos LEDs, controlar a velocidade dos motores CC.

NEGRO: interface analóxica e dual



O módulo **LDR ou SENSOR DE LUZ**, permitiranos encender ou apagar o alumeado en función da luz ambiente

NEGRO: interface analóxica e dual.



AXUSTES DE FÁBRICA

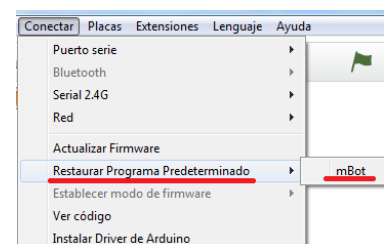
Cando encendemos por primeira vez o noso mBot, observaremos que trae pre-cargados 2 programas diferentes

1. Encende os LEDs de a bordo en cor **BLANCA** , e o robot **estará parado**, en modo infravermellos **-IR -**
2. Se pulsamos o botón da placa mCore, os LEDs cambiarán a cor **VERDE**, e o robot poñerase en modo **“esquiva obstáculos”**, de xeito que xirará aleatoriamente a esquerda ou dereita, cando detecte un obstáculo co sensor ultrasónico a menos de certa distancia.
3. Se voltamos a pulsar o botón, os LED cambian á cor **AZUL**, e o robot ponse en modo **“sigue liña”**, será capaz de seguir unha liña negra sobre un fondo branco (trae unha plantilla).
4. Pulsando de novo , deterase o programa



Cando subamos algún programa empregando o “modo arduino” estes programas pre-cargados desaparecerán.

*Para recuperalos deberemos “**Restaurar o programa predeterminado do mBot**” dende o menú **Conectar***



PROGRAMACIÓN CON mBlock



Apareceranos a icona no escritorio e tamén no menú de aplicacións de Windows.

O programa aparecerá en inglés, podemos cambiarlle o idioma na pestana

“linguaxe” a Galego ou Castelá.

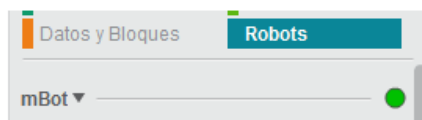
1. Encendemos o mBot

2. Conectamos a placa a un porto USB, comprobamos o porto ao que se conecta (COM...)

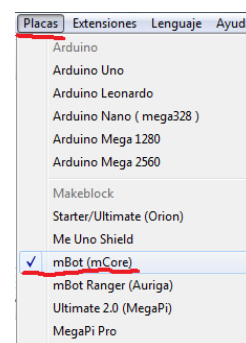
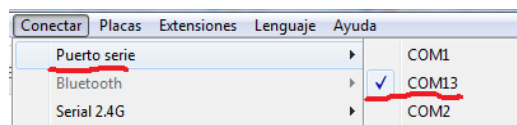
a través do menú **Conectar – Porto Serie**. Se os drivers do mBot están instalados, habilita un porto de comunicación COM novo. Este driver pode instalarse dende o programa mBlock a través do menú **Conectar –Instalar Driver de Arduino**

3. Comprobamos si detecta a tarxeta a través do menú **Placas - mBot (mCore)**

4. *Se o robot está conectado aparecerá un círculo verde nos bloques de programación.*



5. Extensión dos ficheiros mBlock no PC (*.sb2) ollo nos ordenadores abalar, non abrílos con S4A.



A. Programas Non autónomos;

A placa mCore (mbot) pode estar conectada ao PC, de tres xeitos diferentes:

- A través do cable USB
- Conexión inalámbrica 2.4G
- Conexión Bluetooth

Os programas deben empezar sempre con esta icona.



programa elaborado ao facer clic na bandeira verde do escenario.

O robot realizará o

Para poder traballar deste xeito, debemos asegurarnos que a placa mBot contén o **firmware** (software de baixo nivel que controla os circuitos electrónicos de calquera dispositivo) cargado

B. ProgramasAutónomos: Comezan coa primeira instrución



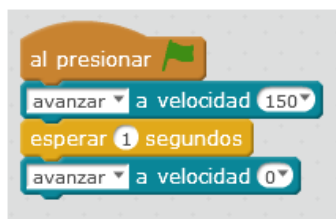
Estes programas **descárganse dende o PC no robot**, mediante o cable USB, facendo clic na icona aparece a pantalla en “modo Arduino” e premeremos no botón

Subir a Arduino

ATENCIÓN: Cada vez que descarguemos un programa na placa a través da opción “**subir a arduino**” e queremos probar outro programa non autónomo, deberemos Actualizar o firmware ou non funcionará.

1. **Actualizar o Firmware**(cando descargamos un programa en modo Arduino e desexamos manexalo de novo “non autónomo” (USB, Bluetooth, 2.4G ou coa APP)
2. Se desexamos que funcione o robot no modo inicial deberemos ir a **Conectar - Restaurar o programa predeterminado**.

Práctica 1: Avanzar , Retroceder e xirar a esquerda e dereita.

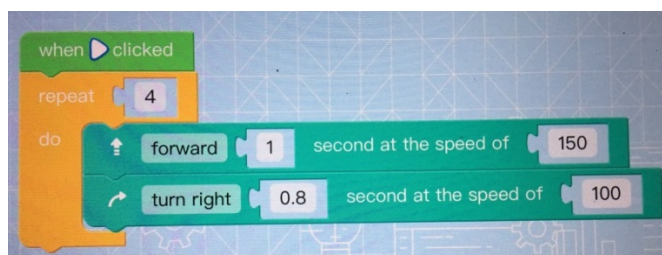


Unha volta completa de roda (1 rotación) =20 cm a velocidade 150 durante 1 segundo. (orientativo, xa que dependerá do tipo de pavimento sobre o que se mova).

Xiros; se queremos xirar 90º farémolo a velocidade 100 durante 1 segundo (orientativo)



Agora deseñaremos un programa que faga avanzar o noso mBot percorrendo un **cajado** (avanza, xira, avanza ,xira...) **Así quedaría o programa para o mBlock na tablet ou móbil, empregando a opción “repetir”** Neste caso non hai que indicarlle que pare os motores, estes avanzan o tempo indicado nos bloques respectivos. Proba a programar un rectángulo, triángulo, pentágono, hexágono.... Un bo xeito de traballar a xeometría, polígonos, ángulos...



Práctica 1.2.-Engadelle as luces dun coche de policía

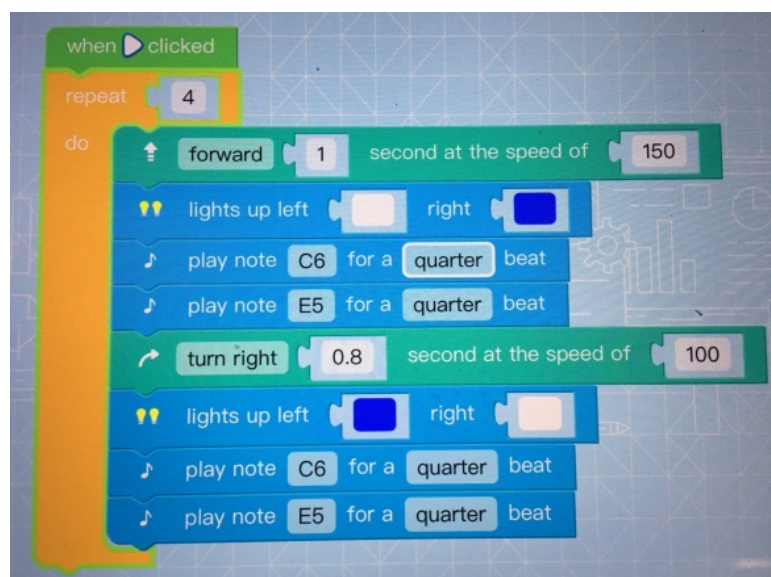
Co seguinte programa (mBlock para PC) ademais de avanzar en cadrado, encenderá os LEDs alternativamente en azul e branco, semellando as luces dun coche de policía.



Práctica 1.3 .- Engádelle a continuación o son da serea.



As notas musicais van do C,D,E,F,G,A,B,(Do, Re, Mi, Fa, Sol, La, Si) e pódese escoller a duración entre: octavo, cuarto, medio, enteiro e dobre (semicorchea, corchea, negra, branca, redonda)



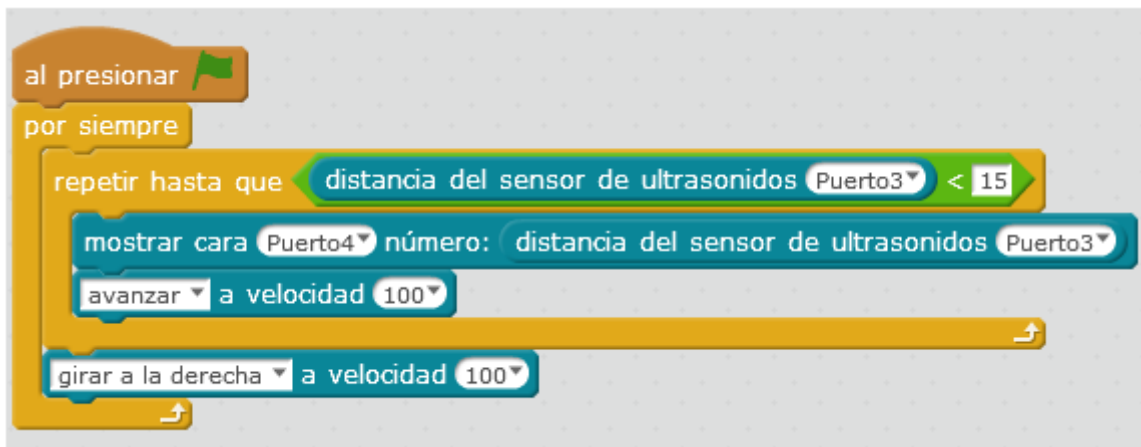
Práctica 1.3.- Esquivar obstáculos . Contedor do lixo no camiño do mBot.

Nesta practica aprenderemos a manexar o sensor de ultrasóns. Este sensor mide a distancia do mBot a un obstáculo en centímetros. Cando atope un obstáculo a menos de 15 cm, entón deterase.

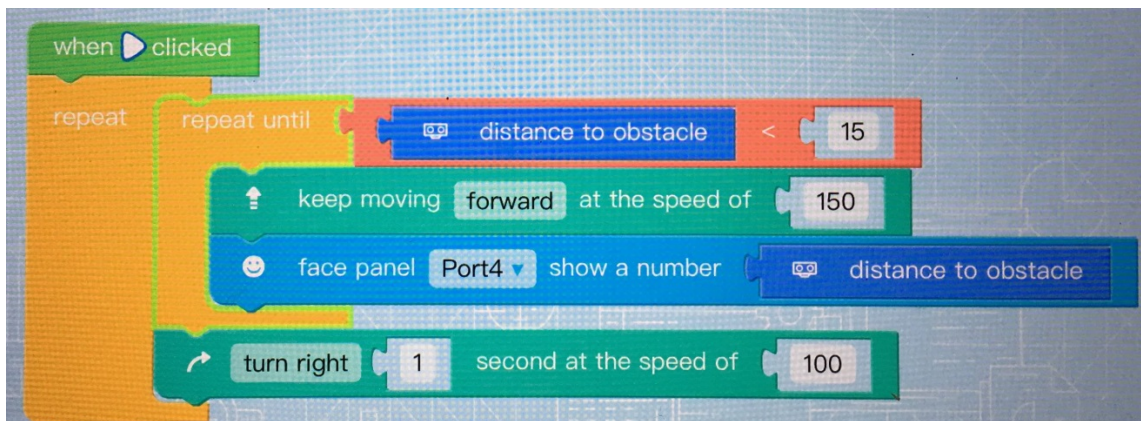


Aquí aparecen dous novos bloques (repetir hasta que... e un bloque matemático “menor que”)

Podemos facer que en lugar de deterse, xire á dereita **“sempre que ”** atope un obstáculo, e ademais nos mostre na matriz de LEDs a distancia ao sensor ultrasónico.



Na aplicación da tablet ou móbil quedaría así:



Práctica 2.2.- Semáforo en mBlock para PC

Nesta práctica pretendemos simular a iluminación dun semáforo empregando os LEDs da placa mCore (a bordo)

As diferentes cores en formato dixital fórmanse a partires de 3 cores RGB (red-green-blue), cuxos valores poden variar dende o 0 ata o 255, combinando diferentes valores para cada un podemos obter unha gran variedade de cores.

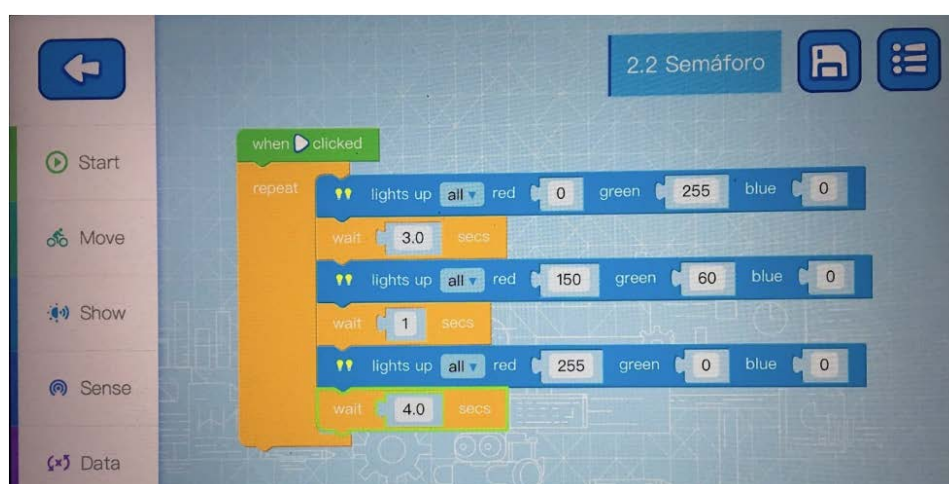
O branco e a mestura de todas as cores (255,255,255)

Para apagar os LED debemos poñelos a cero, senón permanecerán na ultima cor(0,0,0)

[A través do seguinte enlace , podemos ver os valores RGB das diferentes cores](#)



En mBot para Tablet ou móbil (no meu Ipad a versión está en inglés, mentres que no móbil está en castelá)



Práctica 2.3.- Cruce de semáforos

Con esta práctica pretendemos simular o cruce de semáforos que regula o tráfico de dúas rúas. **Empregaremos os dous módulos externos de 4 LEDs RGB**, un deles conectado ao **porto 1** e o outro ao **porto 4**. Se vos fixades estes módulos teñen o **ID** cor **AMARELO**, podemos conectalos a **calquera porto 1,2,3 ou 4**

Haberá que representar dous semáforos que se coordinen;

1. O primeiro semáforo está en verde, o segundo estará en vermello.
2. O semáforo 1 vaise poñendo laranxa.
3. O semáforo 1 poñeráse vermello e o semáforo 2 verde.
4. O semáforo 2 poñeráse laranxa.



Así é como se ve na aplicación para o móbil



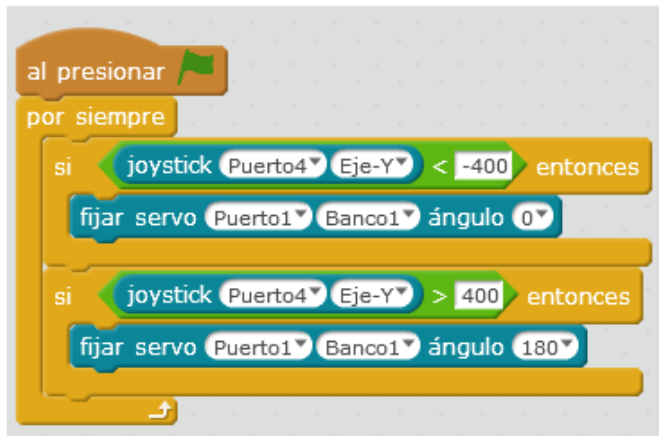
Práctica 3.4.- Controlar o movemento do Servo co Joystick

Un servomotor, é un motor de corrente continua (CC) especial que nos permite controlar a posición do seu eixe de xiro, Está deseñado para moverse uns determinados graos e manterse en dita posición. (este bloque do joystick non está dispoñible na APP)

1. Conectaremos o servo no "slot1" a través dun adaptador RJ25 ao porto 1
2. Conectamos o Joystick ao porto 4

Consiste en que o robot lee os movementos que facemos no Joystick nos eixes X e Y, en función dos valores que detecta no joystick moverá o servo.

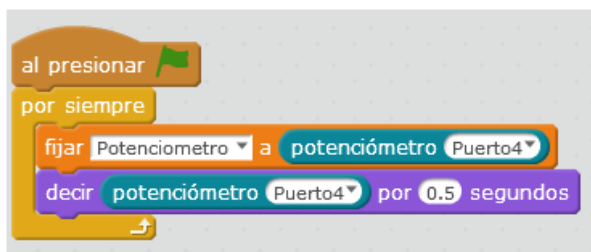
Cando o eixe Y do joystick teña un valor inferior a -400, o servo irá a posición 0º, e cando o valor detectado sexa >400 o servo moverase á posición 180º.



Práctica 3.5.- Controlar o servo con un potenciómetro

Un potenciómetro é unha resistencia á que se lle pode variar o seu valor con un mando que xira. Este potenciómetro facilita uns valores dixitais de entrada entre 0 e 1024. (o rango pode variar, no noso caso varía de 0 a 975)

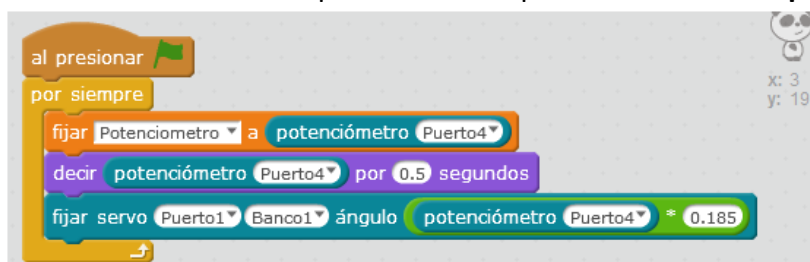
A primeira proba que imos facer será: mostrar o valor que lee no potenciómetro e que o diga o personaxe.



O servo, pode moverse nun rango de ángulos entre 0º e 180º.

Podemos calcular a relación entre os valores do potenciómetro e os valores dos ángulos do servo. A relación entre o ángulo e o potenciómetro será $= 180^\circ / 975 = 0,185$

Así que a posición do motor será a lectura do potenciómetro por a **relación = lect pot * 0,185**



Poderíamos controlar o brillo dos LEDs co potenciómetro?

Práctica 3.6.- Mostrar unha imaxe na matriz de LEDs

Se non a tedes montada, deberedes montar a matriz de LEDs e conectarlle ao porto 4. A continuación actualizaremos o Firmware.

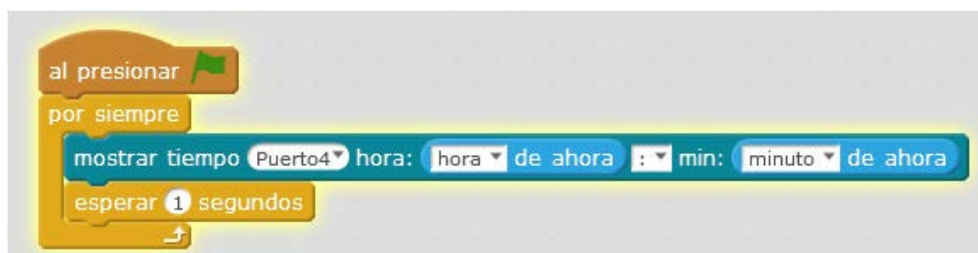
A matriz de LEDs é unha pantalla formada por 128 lámpadas LED, formando unha matriz de 8 filas x 16 columnas. Cada LED pode encenderse e apagarse por separado, e pode mostrar números, letras ou debuxos en función dos LED que encendamos ou apagamos.



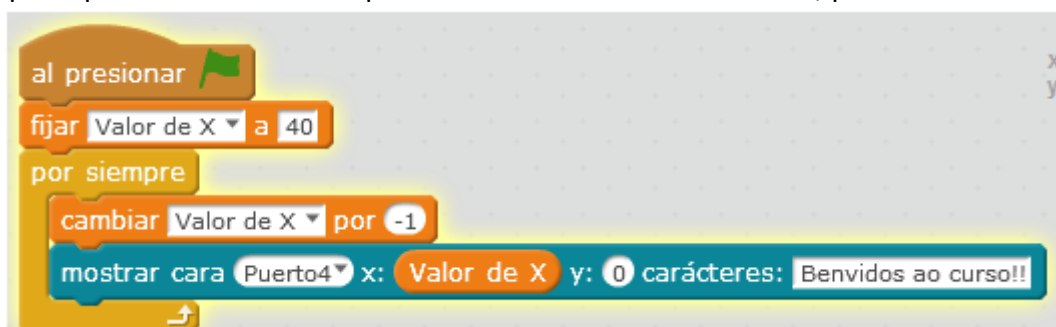
Pode mostrar textos curto



E como non, a hora actual e minutos



Tamén podemos poñer textos longos, para o cal, necesitaremos engadir unha variable “**valor de X**” para que vaia cambiando a posición do texto na coordenada X, pois senón non cabe.



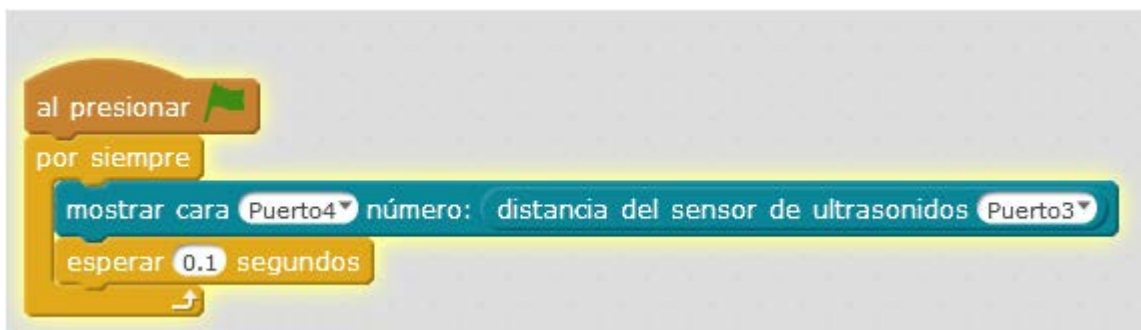
E tamén números, e números romanos



Se queremos que o texto **se repita continuamente** e ademáis o texto non cabe na matriz de LEDs, empregaremos o comando “por siempre”. Neste caso diremos que o texto terá unha lonxitude de 40 e para que se desplace, a coordenada X irá diminuindo ata a coordenada -40, nese momento repítese de novo o bucle. (o valor de longo de X , variará en función da lonxitude do texto)



A matriz de LEDs tamén nos pode mostrar a distancia que está medindo o sensor de ultrasóns a un obstáculo.



Práctica 3.7.- Mando a distancia e sensores IR (combinado coa matriz de LEDs)

O sensor IR situado na parte dianteira da placa mCore, pode recibir do mando números, letras do A ao D, flechas e o botón de configuración.



O mando a distancia, non é exclusivo dun robot, é dicir, os demais robots da aula reciben a mesma información, polo que é importante, apuntar co mando aos sensores do robot co que estamos a traballar, e non a outro.

Podemos controlar o mBot co mando a distancia, coma un coche teledirixido. Coas teclas de dirección avanza, retrocede e xira a esquerda e dereita



Os programas que realicemos co mando a distancia, non poden ser “non autónomos” e dicir, non podemos empregar o comando  e ter conectado o mBot ao noso PC mediante o cable USB.

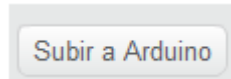
Isto ocorre, por que o mBot ten instalado o programa por defecto (*firmware*) de ler o mando , e non podemos obviarlo.

Así debemos comezar os programas con:

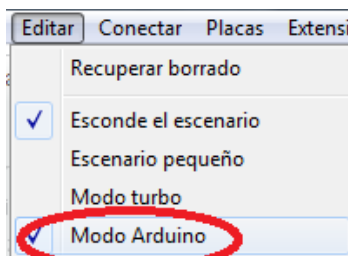
 Programa de mBot

E teremos que cargar a

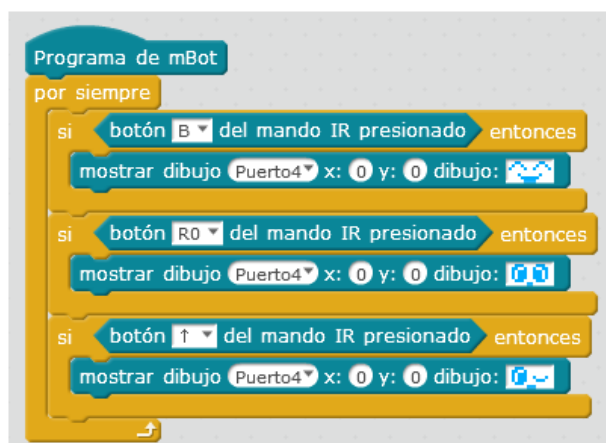
programación no robot co comando “subir a Arduino”

 Subir a Arduino

Para cambiar a “modo arduino” debemos ir ao menú Editar e activar a casilla “modo arduino”



Exemplo 1: Cambiar de cara na matriz de LEDs ao presionar diferentes botóns do mando IR.



As caras permanecerán na matriz de LEDs ata que se preme outro botón.

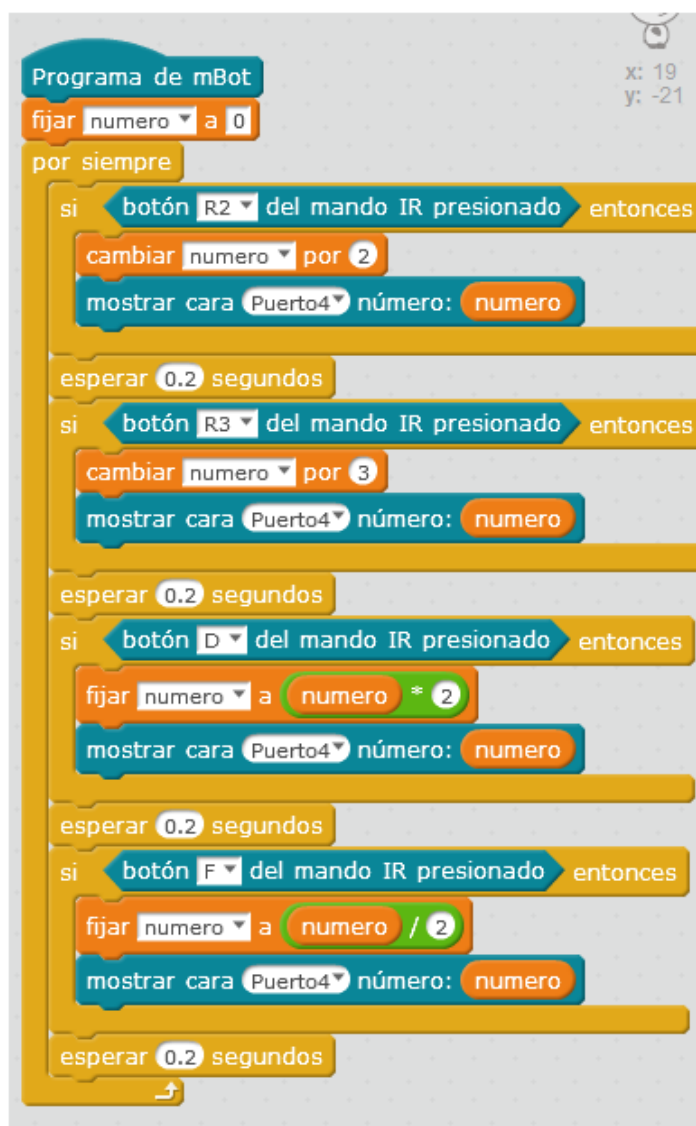
Exemplo2: Sumar e restar números.



Neste exemplo o robot sumará 1 cando pulsemos ↑ e restará 1 cando pulsemos ↓. A espera de 0,2 segundos serve para que con unha soa pulsación do mando cambie só un número de cada vez.

Poderíamos empregalo para que aparezan os números de 2 en 2, de 3 en 3, ou para traballar o cálculo mental.

Exemplo3: Operacións matemáticas



Neste exemplo ao premer:

- R2 sumamos 2
- R3 sumamos 3
- D multiplicamos por 2
- F dividimos por 2

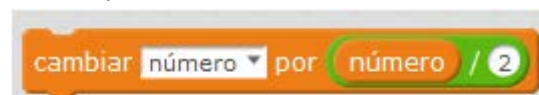
E todas as posibilidades que se nos poidan ocorrer coas operacións matemáticas básicas.

Múltiplos de 3, divisores, números primos, números aleatorios...

Como podedes observar, o bloque

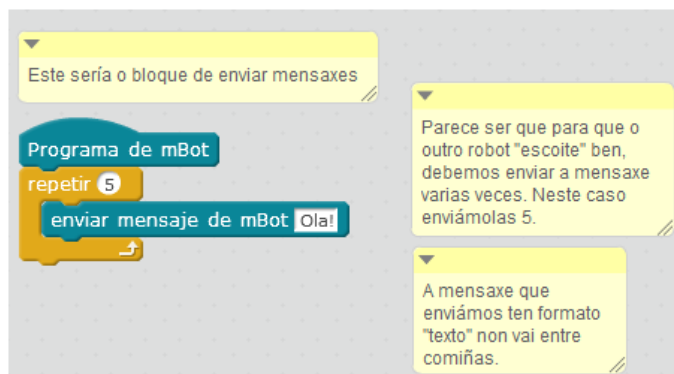
número é unha **variable**, nós podemos crear cantas variables desexemos.

O bloque **cambiar número por 1** pode incrementar ou diminuir a variable que nós poñamos no círculo (neste caso 1), tamén podemos poñer un valor resusultante dunha operación.(dividir entre 2)

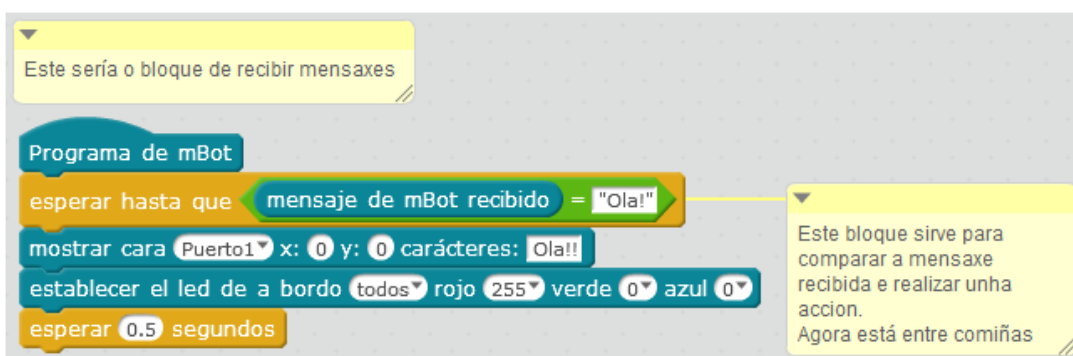


Exemplo 4: Wasapeando

A placa mCore, está equipada cun **receptor** de infravermellos IR, que lle permite recibir ordes do mando a distancia, (tamén de calquera outro emisor de infravermellos, coma o mando do TV ou o móbil).



Esta placa tamén posúe un emisor , que podemos empregar para enviar mensaxes dun mBot a outro. Este emisor está tamén na parte dianteira da placa, así que os robots deberán “mirarse”, en liña recta pode ter un alcance de ata 2 m.



Para ampliar:

1. Escribir 2 programas, de xeito que o emisor envíe unha mensaxe ao chou e o receptor encenda os LED de a bordo , en verde ou vermello en función da mensaxe que recibe.
2. Escribir 2 programas de xeito que un robot envía a información sobre a distancia que o sensor ultrasónico detecta, e o robot que recibe a mensaxe , axusta a súa distancia en consecuencia.
3. Escribir un programa no que 2 robots sigan unha liña negra. O primeiro robot ten un sensor ultrasónico e detense ao detectar un obstáculo a menos de 15 cm, e indicalle ao segundo robot que tamén debe parar.

Práctica 3.8 Sensor de Luz

Esta placa mCore dispón dun **LDR (Light Depended Resistor)** – Resistencia que depende da luz- que podemos empregar para detectar a luz ambiente. Se empregamos o sensor de luz externo , como o sinal que recibe é **ANALÓXICO**, debe conectarse a un porto co **ID** cor **NEGRA**, no noso caso os portos 3 e 4.

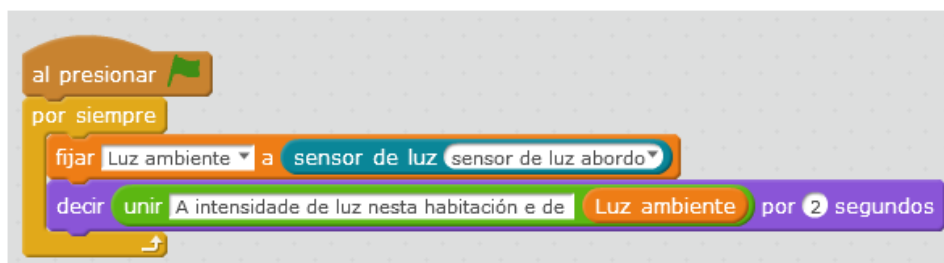
Neste caso pode ser un programa “**non autónomo**” así que podemos executalo conectado ao porto USB e podemos empregar o comando

al presionar

Necesitamos crear unha variable (valor que cambia)

Luz ambiente

Exemplo 1: Nun comezo crearemos un programa sinxelo que nos permita mostrar en pantalla o valor numérico da luz ambiente na nosa habitación, que será medida polo sensor de luz de a bordo

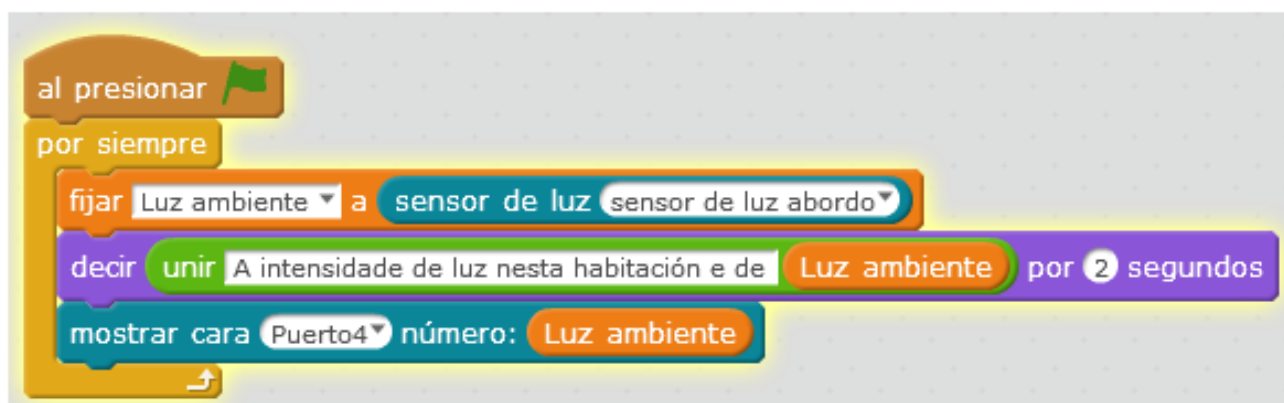


A intensidade de luz nesta habitación e de 287

O persoaxe (obxecto) do escenario, mostrará o valor da luz ambiente.

Tamén se pode mostrar o valor da luz ambiente na matriz de LEDs.

O bloque “**por sempre**” empregámolo para que esteña realizando a medida continuamente.



Exemplo 2: Despertador solar

Cando o sensor de luz de a bordo detecte valores por riba de 800 (valor estimativo) , emitirá unha nota musical.

Engadimos o bloque “condición” **SI... ENTONCES**.

Empregamos tamén o bloque “**por siempre**” para que realice medicións da luz ambiente cada 0,5 segundos (podemos establecer outra espera, que será o retardo en volver a realizar a medición), sen este bloque realizaría unha única medición, e necesitamos que comprobe o valor do sensor continuamente.

Tamén lle pedimos que mostre o valor do sensor na matriz de LEDs.



Exemplo 3: Encender o alumado público.

Empregaremos:

- 2 módulos externos de LEDs, que serán o noso alumado público (farolas), conectados nos portos 1 e 4
- O sensor LDR de a bordo
- O bloque “**SI..... ENTONCES.....SINÓ**”

Deste xeito que cando a iluminación sexa menor de 200, acenderá o alumado e cando o valor medido polo sensor supere ese valor, apagará os LEDs.

Neste exemplo, a espera entre medicións é de 5 segundos.



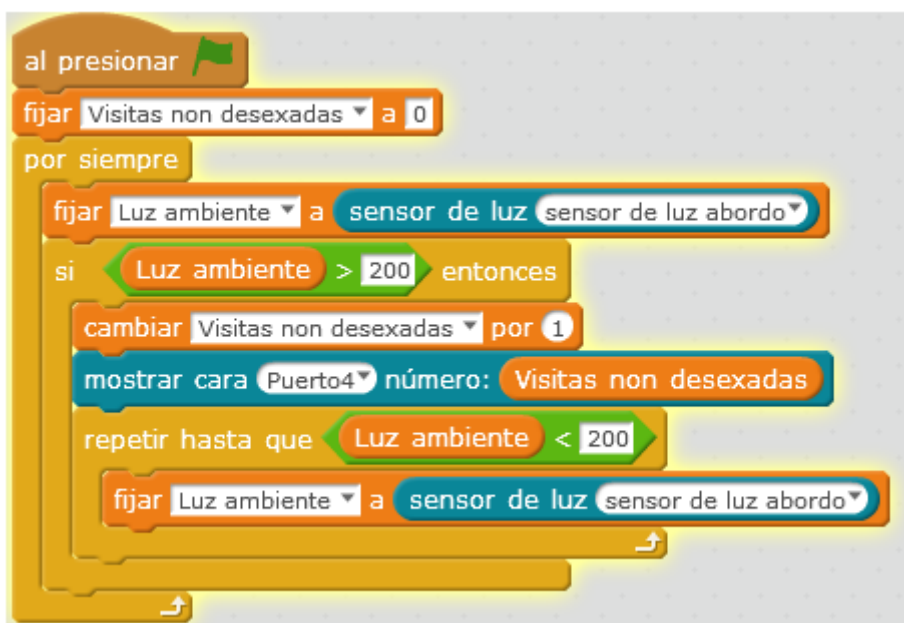
Enlace ao video en youtube - <https://youtu.be/yLNiIP3PZbc>

Exemplo 4: Un programa que **detecte as visitas non desexadas** no meu cuarto , cando non estou.



Ollade as dúas variables; “visitas non desexadas” e “ luz ambiente”. Cada vez que aumenta a luz por riba de 200, conta unha visita máis.... pero, este programa ten un pequeno erro, atopas onde?

O bucle “por siempre” fai que sume as visitas continuamente mentres a luz está encendida.



O correcto será que sume unha visita e espere a que se apague a luz, esperando a seguinte visita.

programa deberá ser a

Outro exemplo de detector de intrusos pódese facer co **SENSOR PIR**, neste caso, cando detecta un intruso (movemento) reproduce un son e mostra unha mensaxe na matriz de LEDs.

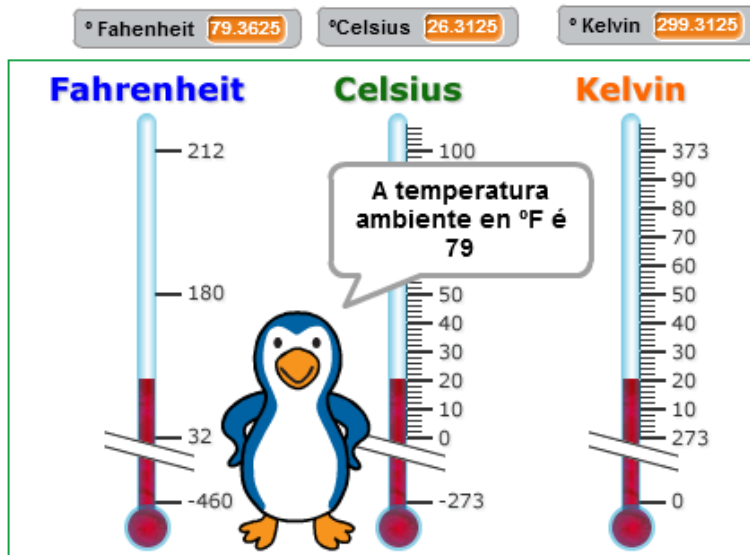
É importante lembrar que a distancia a detectar polo sensor pódese axustar nunha pequena resistencia variable que contén o módulo.



Práctica 3.9 Temperatura do café

Trátase de programar o robot para que mida a temperatura ambiente (ou dun café) en °Celsius, °Kelvin e °Fahrenheit. Neste caso que a mostre na pantalla do Scratch e que a diga o Pingüín

O rango de temperatura que mide esta sonda está entre -55°C e +125°C



Para convertir °Celsius a °Kelvin
 $^{\circ}K = ^{\circ}C + 273$

Para convertir °Kelvin a °Celsius
 $^{\circ}C = ^{\circ}K - 273$

Para pasar de °C a °F
 $^{\circ}F = (9/5 * ^{\circ}C) + 32$

Para pasar de °Fahrenheit a °Celsius
 $^{\circ}C = (^{\circ}F - 32) * 5/9$

Si desexamos que mostre a temperatura no display de 7 segmentos e 4 números, conectado ao porto 1;

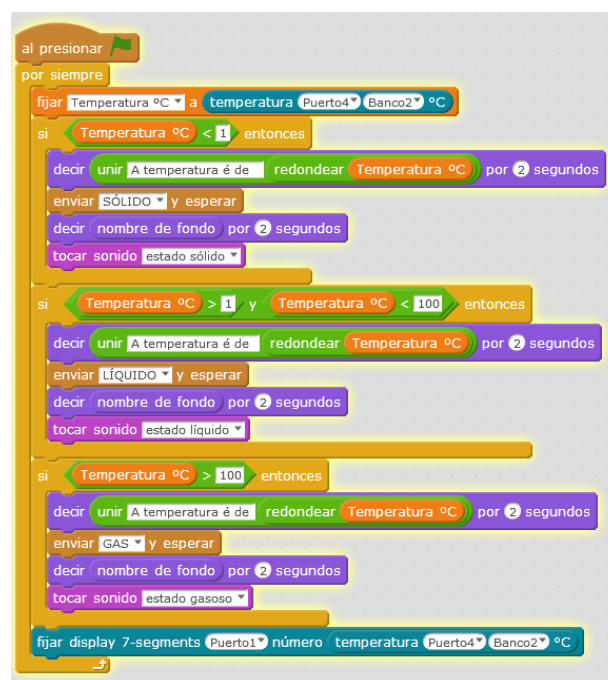
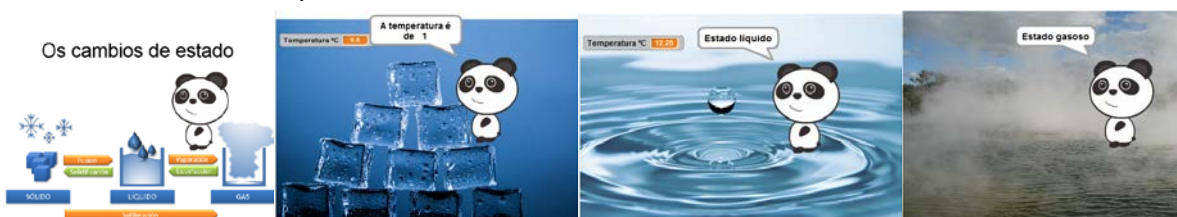
1. Conectamos o display de 7 segmentos no porto 1
2. Conectamos a sonda de temperatura no conector 2(slot 2) a un módulo adaptador RJ25
3. Conectamos o módulo adaptador con un cable RJ25 a o porto 4 da placa mCore do mBot

Práctica 3.10.- Cambios de estado da auga

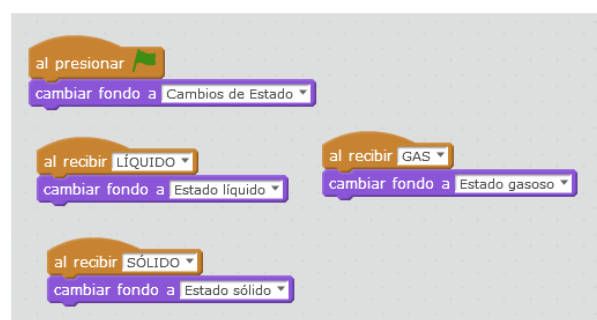
Trátase de programar o mBot para que mida a temperatura da auga nos 3 estados, e nos mostre o resultado en °Celsius, indique o estado no que se atopa e cambie o escenario no programa mBlock. O rango de temperatura que mide esta sonda está entre -55°C e +125°C.

Empregaremos;

- A placa mCore
- A sonda ou sensor de temperatura conectada ao Slot2 dun adaptador para portos RJ25 conectado ao porto 4.
- Un display de 7 segmentos e 4 números conectado ao porto 1.
- 4 escenarios
- 1 variable “temperatura °C”



Esta sería a programación para os diferentes escenarios



Práctica 4.1.- Segue liña negra

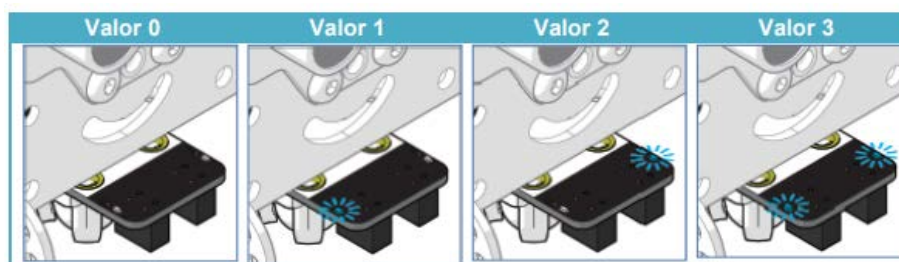
Para realizar este programa empregaremos o sensor de sigue liña, que está situado na parte frontal do mBot e conectaremos ao **porto 2** da placa mCore.

Este módulo está deseñado para detectar unha **liña negra**. Consta de **dous sensores (sensor 1= esquerdo, e sensor 2= dereito)** con 4 elementos, dous LED IR que emiten luz, e dous foto transistores IR que reciben o sinal (reflicte todo =branco ou non reflicte nada=negro).

Grazas a eles, podemos saber se está sobre unha liña negra sobre fondo branco e poder seguila ou sobre unha liña branca sobre fondo negro.

Á hora de programar , podemos atopar 4 posibilidades.

1. **Valor 0**; indica que ambos sensores (**sensor 1 e sensor 2**) están **totalmente sobre a liña negra**.
2. Valor 1, indica que o **sensor esquerdo (sensor 1)** está sobre a **liña branca** e o sensor 2 sobre a cor negra.
3. Valor 2. Indica que o **sensor dereito (sensor 2)** está sobre a **liña branca** e o sensor 1 sobre a cor negra.
4. **Valor 3**, indicará que **ambos sensores** están totalmente sobre a **liña branca**.



O programa que debemos realizar para que **siga unha liña negra** será o seguinte:

- Cando ambos sensores están sobre a liña negra, avanza. (valor 0)
- Se o sensor dereito está sobre o branco e o esquerdo sobre negro, deberá xirar cara á esquerda.(Valor 1)
- Se o sensor dereito está sobre o negro e o esquerdo sobre branco, deberá xirar cara á dereita.(Valor 2)
- Se ambos están sobre branco, retrocederá.(Valor 3)



Práctica 4.2.- Segue liña branca

O programa que debemos realizar para que **siga unha liña branca** será o seguinte:

- Cando ambos sensores están sobre a liña negra, retrocede. (valor 0)
- Se o sensor dereito está sobre o branco e o esquerdo sobre negro, deberá xirar cara á dereita.(Valor 1)
- Se o sensor dereito está sobre o negro e o esquerdo sobre branco, deberá xirar cara á esquerda.(Valor 2)
- Se ambos están sobre branco, avanzará.(Valor 3)

