

MANUAL KIT ARDUINO

Proba práctica



I Olimpiada da Enerxía

Liña temática 2

Producción de enerxía eléctrica

TEMA 1: PRESENTACIÓN ARDUINO

- PROFESORA** Este curso imos aprender a usar unha ferramenta súper útil e divertida. ARDUINO
- ALUMNADO** ¿Qué é Arduino?
- PROFESORA** Arduino é unha plataforma deseñada para facilitar a creación de proxectos electrónicos interactivos
- ALUMNADO** ¿Ein? ¿Que demo quere dicir iso?
- PROFESORA** É un equipo electrónico co que poderemos medir temperaturas, son... e controlar, por exemplo, luces, motores eléctricos... Soa ben, ¿non si?
- ALUMNADO** ...De tolos... ¿E como é iso posible?
- PROFESORA** Vedes esta plaquiña tan chula que teño aquí. Pois o corazón do Arduino é este chip. Chámase microcontrolador e, basicamente, é coma un ordenador. Podemos programalo para que faga o que queiramos.
- ALUMNADO** Ahhh, e ¿por que non podemos usar directamente un ordenador normal?
- PROFESORA** ¿Vedes estes buratiños de aquí? Aquí podemos conectar uns cables dos sensores que queiramos usar e ler os seus sinais directamente. ¿A que o voso ordenador da casa non ten iso?
- ALUMNADO** Non
- PROFESORA** ¿Vedes estas outras conexións de aquí? Pois esas son saídas e serven para controlar actuadores como, por exemplo, as luces da casa, subir unha persiana, acender unha alarma...
- ALUMNADO** Eh, pero esas cousas xa as pode facer SIRI por min...
- PROFESORA** Pois si, é algo parecido, pero SIRI ou o teu móbil só poden facer algunhas cousas para as que están programados. Con Arduino podedes crear as vosas propias solucións.
- ALUMNADO** Soa interesante, aínda que ten pinta de que ese aparello é bastante caro...
- PROFESORA** Pois non. Esa é outra das grandes vantaxes de Arduino. É súper económico. Esta placa custa só 30 € e atoparemos un montón de sensores e controladores de baixo custo para facer case calquera cousa que imaxinemos...
- ALUMNADO** ola... A ver canto pesa o trebello ese...

1. Conceptos básicos

Para que serve un Arduino?

TEMA 2: INTRODUCCIÓN HARDWARE

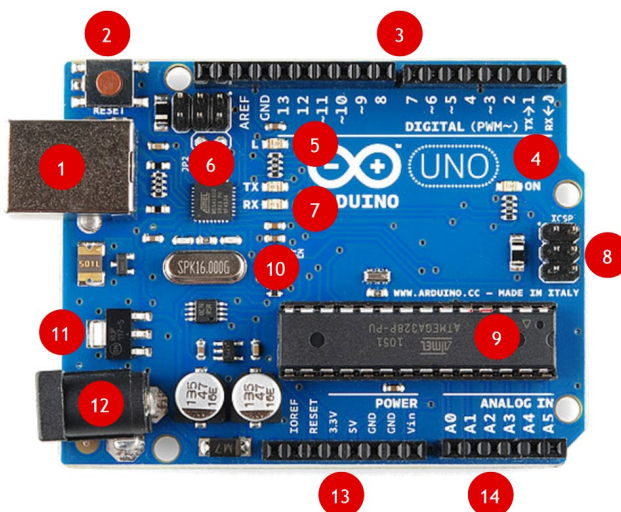
PROFESORA Hoxe imos comenzar a familiarizarnos cos nosos equipos. Collede vosa placa de Arduino ONE® pola parte de diante...

ALUMNADO O temos!!

PROFESORA Moi ben, o que tendes diante é una placa electrónica. Para os que non lembredes o que é iso, podedes ver este vídeo: [VIDEO PCB](#)

ALUMNADO Si, lembrámolo. Chámase PCB e é una lámina que serve para conectar compoñentes electrónicos de forma organizada. Cos camiñitos eses para a electricidade...

PROFESORA Exacto. Non son camiñitos, senón pistas, pero ben. Pois imos ver as partes miás importantes desta placa. O cerebro do Arduino (9) é un procesador que podemos programar para que faga o que queremos.



ALUMNADO E como nos comunicamos con el?

PROFESORA Pois imos vinculalo con oso ordenador usando un cable USB que se conecta en (1), así que tamén necesitamos un ordenador para traballar.

ALUMNADO Pero, non se supón que isto xa era un ordenador? Non é capaz de funcionar por sí só?

PROFESORA Claro que funcionará por sí só, pero primeiro teremos que poñelo a andar. Hai que crear un programa para que despois faga o que nós queiramos. E para iso necesitamos un ordenador.

ALUMNADO Ah, ok...

PROFESORA Moi ben, as outras partes importantes que debes coñecer son: a conexión de alimentación eléctrica (12), conexións de subministro eléctrico (13), entradas analóxicas (14) e as entradas e saídas dixitais (3).

ALUMNADO Perdoa, profe, un pouco máis despacio. Para que conectamos a placa?

PROFESORA Todos os equipos electrónicos consumen enerxía polo que debemos alimentalos con electricidade para que funcionen. Este Arduino necesitará unha tensión entre 7-12V, polo que o conectamos a un transformador, como o que usamos para cargar os móbiles.

ALUMNADO Pero cunha conexión diferente, non?

PROFESORA Pois si, pero a función é a mesma. Unha vez que a placa está alimentada, podemos facer que ela mesma alimente eléctricamente outros

circuítos, coas saídas (13) que comentabamos. Vedes que podemos dar 3.3V ou 5V...

ALUMNADO O do voltaxe era aquilo da altura do depósito de auga?

PROFESORA Si. Os circuítos eléctricos pódense entender como 2 depósitos de auga conectados cunha tubaría. Lembrades? Se ambos están á mesma altura, non circula a auga. É dicir, non habería intensidade eléctrica na instalación se non hai diferenza de potencial eléctrico. Pero no momento en que elevamos un dos depósitos, a auga comeza a fluir. Iso sería como elevar a tensión nun dos extremos do circuítos. E canto máis alta sexa esta tensión, máis auga circula...

ALUMNADO Xa lembro. E cando máis pequena a tubaría, entón máis resistencia e menor caudal de auga.

PROFESORA Exacto. No circuítos eléctrico, canto menos resistencia, máis intensidade ou corrente circulará. Que ben o explicaba o vídeo de [CIENCIABIT](#) É moi importante que recordedes a [LEY DE OHM](#)

$$V = I \cdot R \quad I = V / R \quad R = V / I$$



ALUMNADO Entón, nestes oquediños poderemos conectar cables e usar a súa tensión. E o que pon GND, que voltaxe ten?

PROFESORA Eses oquediños chámanse pines e ese xusto non ten tensión. GND é a abreviatura de Ground, en galego terra. Lembrade cando a usábamnos como sistema de seguridade no caso de que haxa algún problema.

ALUMNADO O vídeo do [CABLE DE TIERRA](#) que me custaba entender...

PROFESORA Pois si. Pero aquí imos usar ese terminal non como terra, senón como a nosa referencia de tensión. É dicir, 0 Voltios. Segue todo o mundo ata agora?

ALUMNADO Por agora todo controlado.

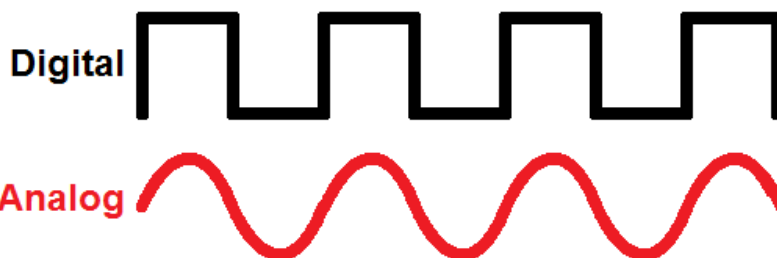
PROFESORA Xenial. Pois sigamos cos pines de entradas. Os pines 3 e 14 que poñen INPUT serven como entrada de sinais.

ALUMNADO Sinais? De tráfico? Que é iso?

PROFESORA Unha sinal é a variación de algunha magnitude física. Por exemplo, a temperatura medida cun termómetro. É dicir, cando conectemos sensores, estes enviarán a información de aquilo que miden ao Arduino a través destas conexións. Pero uns farano con sinais dixitais (3) e outros con sinais analóxicos (14). Sabedes a diferenza?

ALUMNADO Iso soa a que unhas son máis antigas ca outras ou algo así, non?

PROFESORA Ás veces usáanse con ese sentido, pero en realidade é algo máis complicado. Ambas as sinais son usadas no mundo moderno e son todas igual de necesarias. Para entender a súa diferenza necesito que vexades con calma este vídeo de [DATOS ANALÓGICOS E DIXITAIS](#)



- ALUMNADO** Entón, as analóxicas poden tomar infinitos valores, mentres que a dixital só ten dous (ceros e uns). Entón, a dixital é mellor para sinais que só teñan 2 estados, como encendido/apagado?.
- PROFESORA** Pois si, aínda que tamén existe a posibilidade de converter analóxicas en dixitais e viceversa. Pero por agora non imos complicar tanto o tema.
- ALUMNADO** Por que as sinais dixitais do Arduino non poñen INPUT?
- PROFESORA** Porque a diferenza das analóxicas, que son só de entrada, as dixitais son entradas e saídas. É dicir, serven para ler información, pero tamén para controlar actuadores.
- ALUMNADO** Que cutre este equipo sen saídas analóxicas...
- PROFESORA** En realidade, si que ten. Se fixádesvos ben, hai algúns pines das saídas dixitais que teñen este símbolo (~). Iso significa que se poden usar para control analóxico a través dunha técnica que se denomina PWM, pero xa falaremos diso máis adiante.
Non pretendades correr antes de saber andar, pequenos saltamontes 😊
- ALUMNADO** Jajajaja. Vale, vale... con toda a información de hoxe xa me vai chegando...
- PROFESORA** Xa se vai facendo tarde, si. Por sorte, xa rematamos coa introdución. Se alguén se perdeu coa explicación de hoxe, déixovos aquí un vídeo no que podedes repasar as partes fundamentais do Arduino ([INTRO A ARDUINO](#)).
Mañá faremos o noso primeiro programa!!

2. Conceptos básicos

Que é unha PCB?

Cales son os elementos principais dunha placa Arduino?

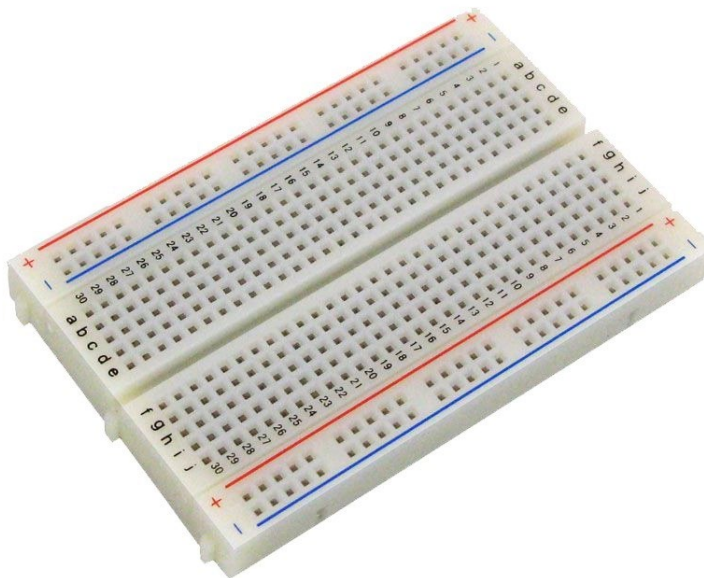
Lei de Ohm e circuítos eléctricos básicos

Que é unha sinal?

Tipos de sinais. Analóxicos vs dixitais

TEMA 3: PROTOBOARD E PRIMEIRO CIRCUITO

PROFESORA Bos días! Vos está gustando esta clase de tecnoloxía? Se é así, imos seguir. Hoxe imos crear o noso primeiro circuío. Para iso imos usar unha tarxeta perforada que tedes no voso kit de Arduino. Chámase protoboard ou tarxeta de prototipos.



ALUMNADO Esta branca que está chea de oquiños?

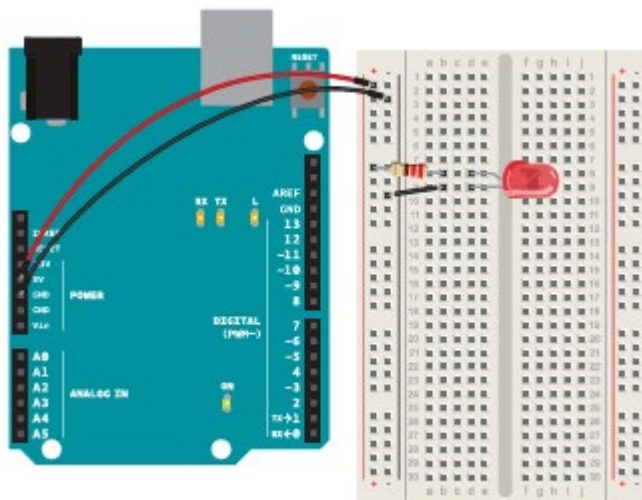
PROFESORA Exacto. Imos aproveitaros para conectar os nosos cables ou conectores nela. Aquí tedes un vídeo explicativo de como funciona a **PROTOBOARD**.

ALUMNADO Entón, as columnas con signos (+ e -) están conectadas verticalmente e as filas centrais horizontalmente. Iso resulta moi cómodo para non ter que poñer moitos cables na placa.

PROFESORA Correcto. Esa é a idea. É unha placa moi útil e a usaremos para poñer a punto diversos circuíos eléctricos mentres aprendemos a usar Arduino.

ALUMNADO Parece sinxelo!

PROFESORA E así é. Para comprobalo, imos facer un circuío que acenda un LED, subministrando a corrente eléctrica desde o propio Arduino. Tal e como se amosa nesta imaxe:



- ALUMNADO** Se algún ten dúbidas, pode seguir este vídeo de [VT EN LÍNEA](#) Fixémolo tal e como indica aí. Iluminou á primeira. Pero hai algunha cousa que non entendo ben.
- PROFESORA** Dime...
- ALUMNADO** Por que é necesario poñer unha resistencia?
- PROFESORA** O LED ten unha resistencia moi, moi baixa. Se o conectamos directamente entre a fonte de tensión e a terra, sería como conectalas directamente. Faríamos un cortocircuito. Sería o mesmo que conectar os 2 bornes dunha batería directamente cun cable...
- ALUMNADO** Ah, xa, claro. E outra cousa. O LED ten dous polos. Pasa algo se os colocamos ao revés no noso circuito?
- PROFESORA** Pois iso é importante. Proba a ver que sucede se o conectas ao revés. Non funcionará. A diferenza da resistencia, que non ten polaridade, o LED debe conectarse no sentido correcto.
- ALUMNADO** Pois o noso non funciona. Está todo igual que no vídeo, pero non se encende o LED...
- PROFESORA** A ver, déixame ver... Xa sei o que pasa. Non tendes conectada a alimentación ao Arduino, e por iso non funciona.
- ALUMNADO** Pero eles tampouco a teñen...
- PROFESORA** Se vos fixades ben, si que a teñen. Está conectada por USB ao ordenador, así que recibe a corrente de alimentación usando ese porto. Lembra que podemos alimentar os nosos circuitos das 2 maneiras...
- ALUMNADO** Ok. Agora xa funciona. Terémolo en conta para as próximas ocasións...

3. Conceptos básicos

Que é unha protoboard ou placa de prototipos?

Coñecer e aplicar a conexión de elementos na placa de prototipos

Manexo dos primeiros compoñentes electrónicos

TEMA 4: SOFTWARE DE ARDUINO

PROFESORA Hoxe imos comezar a familiarizarnos co software para programar Arduino. Podedes descargalo desde a páxina oficial da iniciativa Arduino: <https://www.arduino.cc/en/software>

ALUMNADO Non atopo o enlace onde pulsar...

PROFESORA Á dereita de Arduino IDE, buscade a descarga en función do sistema operativo que teña o voso equipo...

ALUMNADO Ahh, ok, xa está. Pero saltan varios avisos para donar diñeiro, suscribirse a unha revista de novidades...

PROFESORA Non vos preocupedes. Non é necesario que fagades nada diso se non queredes. Simplemente pulsade “Just download” e listo. Agora teremos que executar o arquivo .exe que se descargou e instalar el programa.

ALUMNADO Por que se chama IDE? Que significa iso?

PROFESORA IDE é o acrónimo de Integrated Development Environment. Básicamente unha aplicación que ten ferramentas que facilitan a creación de software ou programas. Non vos preocupedes moito por iso. Non é importante. Se alguén se perdeu, podedes facer todo o proceso de descarga e instalación tal e como se indica neste [VÍDEO DE EXEMPLO](#). Aínda que é mellor que utilicedes unha versión do programa máis actual ca que amosa aí.

ALUMNADO Eu vexo esta pantalla tan rara:

```
sketch_nov18a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

```
Output
Instalando SD@1.3.0
Instalado SD@1.3.0
Descargando Servo@1.2.2
Servo@1.2.2
Instalando Servo@1.2.2
Instalado Servo@1.2.2
Descargando Arduino_BuiltIn@1.0.0
Arduino_BuiltIn@1.0.0
Instalando Arduino_BuiltIn@1.0.0
Instalado Arduino_BuiltIn@1.0.0
Descargando LiquidCrystal@1.0.7
LiquidCrystal@1.0.7
Instalando LiquidCrystal@1.0.7
Instalado LiquidCrystal@1.0.7
```

Estaría todo ben?

PROFESORA Si. O primeiro que imos facer, para que traballedes máis cómodos, é cambiar o idioma. En *file – preferences* - cambiar o idioma. E unha vez feito isto, imos conectar por primeira vez o cable do noso Arduino ao USB

do ordenador. Se todo se instalou ben, debería identificar o equipo aquí:



Se tedes algún problema podedes seguir os pasos que indican neste vídeo de [INSTALACIÓN](#).

- ALUMNADO** Bufff, que largo é ese vídeo... Pero si isto é moi sinxelo...
- PROFESORA** Claro que é fácil, pero nese vídeo xa vos explican algunha cousa de como funciona o programa. É importante que o vexades completo, porque imos traballar algúns deses conceptos. Algúns conceptos importantes quizá se entendan mellor con este vídeo de [JUANRA GARCIA MONTES](#)
- ALUMNADO** Ok...
- PROFESORA** Hai unha cousa moi interesante que debes saber. Se entrades en *archivo - exemplos*, atoparedes algúns códigos xa escritos que nos poden servir para aprender e aforrarnos algo de tempo en proxectos posteriores.
- PROFESORA** Os programas que imos crear en Arduino reciben o nome de sketches ou bocetos. O linguaxe que usaremos baséase en C/C++, pero con algúns cambios para adaptalo e facer a programación máis sinxela, o cal é moi útil. Na pantalla imos escribir as liñas do código, que se irán executando de maneira secuencial (ordenada).
- ALUMNADO** Pero entón, necesitamos aprender ese linguaxe para utilizar Arduino?
- PROFESORA** Si, é necesario coñecelo para ir creando o noso código, pero imos aprender as cousas máis básicas con algúns exemplos nos próximos días. Pouco a pouco. E cando necesitemos algo novo, sempre teremos recursos en internet que nos poidan axudar.
- ALUMNADO** Refírese a pedirlle a outra xente que nos escriba os nosos programas?
- PROFESORA** Ben, algo así. Na internet atoparemos moita axuda e en ocasións algoritmos ou programas completos xa feitos por outra xente que podemos utilizar. Ou adaptar para a aplicación que queremos. Pero temos que coñecer un pouco como funciona para asegurarnos de que fai exactamente o que queremos.
- ALUMNADO** E hai moita xente que traballe con Arduino?
- PROFESORA** Un montón... Esa é a súa gran fortaleza. Arduino forma parte dunha iniciativa de hardware e software libre para a construción de dispositivos dixitais. Iso significa que hai moita xente colaborando gratuitamente no seu desenvolvemento e ofrecendo código e solucións á comunidade de maneira desinteresada e para que creen por si mesmos ([DO IT YOURSELF](#))
- ALUMNADO** Sen gañar nada? Que pringaos...

- Ben, quizais cando crezas un pouco o entendas mellor, pero no mundo hai moitas cousas máis importantes ca o diñeiro ou o interese persoal 😊
- ALUMNADO** ...
- PROFESORA** Tanto é así, que podedes atopar páxinas ou canles de youtube que se dedican por completo a ensinar á xente o uso de plataformas como esta, como:
- [LUIS LLAMAS](#)
 - [CEFUVE ELECTRONICS](#)
- PROFESORA** Nos próximos días comezaremos a crear máis programas para familiarizarnos co IDE e o linguaxe de Arduino. A partir de hoxe quero que teñades sempre á man [ESTA GUÍA](#). A usaremos a miúdo... En especial estas funcións que vos deixo a continuación:
- **pinMode(pin, mode):** Configura un pin como entrada ou saída.
 - **digitalWrite(pin, value):** Envía un valor (HIGH o LOW) a un pin.
 - **digitalRead(pin):** Le o estado de un pin (HIGH o LOW).
- analogWrite(pin, value):** Envía unha sinal analóxica (PWM) a un pin.
analogRead(pin): Le un valor analóxico de un pin (0 a 1023).
delay(ms): Detén a execución por un número de milisegundos.
- ALUMNADO** Oooooook, eso está feito... Ata mañá.

4. Conceptos básicos

Conceptos básicos de programación:

Algoritmo, programa e linguaxe de programación

Estrutura do IDE de Arduino e configuración básica

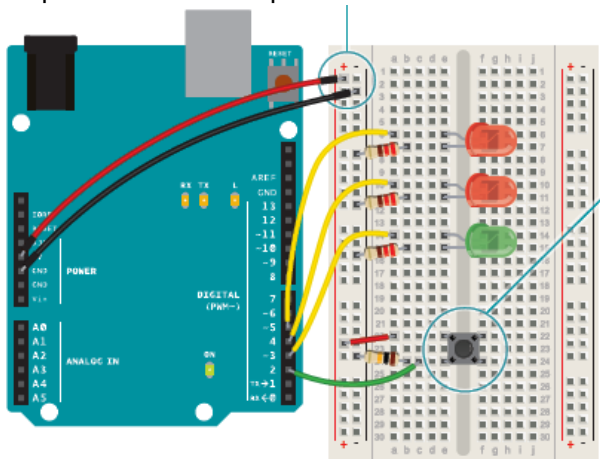
Utilidade das funcións setup e loop

TEMA 5: AS SAÍDAS DIXITÁIS

PROFESORA Bos días chic@s! Hoxe imos facer un proxecto no que traballaremos o control de sinais dixitais de saída co noso Arduino. Lembrades o que significaba dixitais?

ALUMNADO Por suposto. Que só tiña 2 niveis posibles. Acendido (1) ou apagado (0).

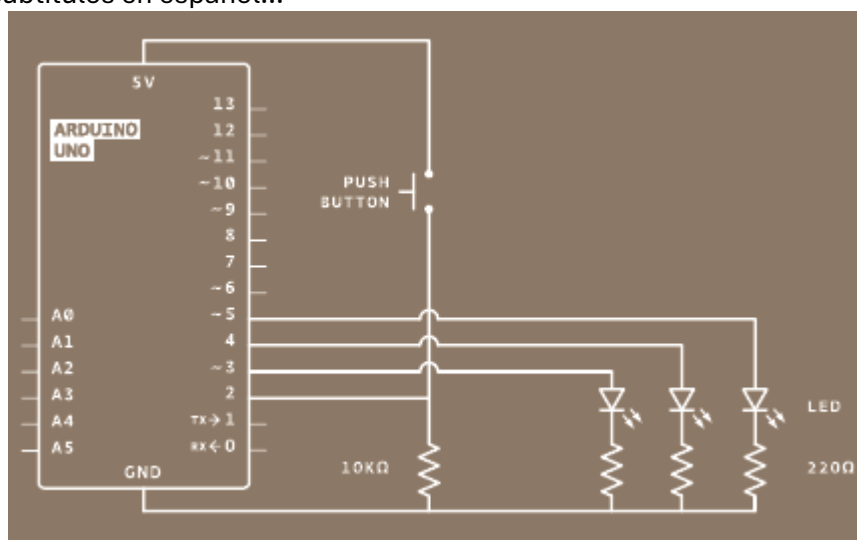
PROFESORA Iso é. Ou para ir familiarizándonos co vocabulario de Arduino, as sinais serán altas (HIGH) ou baixas (LOW). Pois ben, imos construír un montaxe con 3 leds e un pulsador como o que se ve na imaxe:



O obxectivo é que cando o programa arranque, o led verde estea acendido de maneira continua, ata que usemos o pulsador. Mentres estea pulsado, os leds vermellos deben acenderse de maneira alterna, parpadeando continuamente ata que se finalice a secuencia ao soltar o pulsador.

ALUMNADO Uf, canto cable. Isto parece un pouco complexo...

PROFESORA Non vos agobiedes. É ben sinxelo e imos utilizar [ESTE VÍDEO](#) de guía paso a paso para que ninguén se perda. Aínda que está en inglés ten subtítulos en español...



Eu irei axudándovos mentres avanzades...

Lembrade que antes de nada o que debemos facer é pensar na secuencia de procesos que deben suceder e que eventos os

- desencadean. Lembrades os diagramas de fluxo? Pois temos que pensalo dese xeito...
...Alumnado traballando duramente...
- PROFESORA** Lembrade que unha boa práctica cando se programa é engadir aclaracións ou comentarios no código. Pódense introducir poñendo dobre barra (`// Texto`). Isto axudará no futuro a lembrar por qué se fixo o código así.
- PROFESORA** Lembrade tamén que cada liña debe finalizarse con `;` para que sexa executada polo compilador...
- ALUMNADO** Compilador? Que é iso? Eu non teño ningún, creo...
- PROFESORA** Non te preocupes, si que o tes. O compilador é un software que traduce o programa que estamos escribindo a un código máquina que pode entender o microprocesador. Así nós non necesitamos programar usando diferentes linguaxes. É algo así como un traductor e xa está instalado e funcionando no software de Arduino sen que nós fagamos nada. Esa tradución prodúcese cando lle damos a executar ao programa...
- ALUMNADO** Ah, vale... Eu creo que xa temos claro o programa.
- PROFESORA** Perfecto, pois agora montade o circuito eléctrico na placa de prototipos, seguindo o esquema mostrado. Para escoller as resistencias adecuadas podedes lembrar o [CÓDIGO DE CORES](#) que explicamos en teoría de circuitos.
- ALUMNADO** É así?
- PROFESORA** Si, está perfecto. Vou darvos un truquiño para aforrarnos algo de tempo... No IDE ides a *archivo - exemplos - StarterKit_BaseKit - p02_SpaceShipInterface*. Abrid ese arquivo e xa tedes o código que necesitamos. Imos analízalo un pouco...
- ALUMNADO** Ei, xa nos poderías ter dito antes, que levamos un bo rato escribindo código e é bastante lento...
- PROFESORA** Primeiro debemos configurar o uso que lle imos dar aos pines. Iso facémolo no setup **void setup()** e declaramos cales son saídas e entradas co **pinmode**. Tede coidado porque o código distingue entre maiúsculas e minúsculas, e se escribimos mal as funcións non funcionarán.
- ALUMNADO** Ah, ves, iso era o problema que nos pasou a nós ao principio...
- PROFESORA** O comando **int** serve para crear unha variable. Ese nome elíxeno nós e tamén o seu estado inicial. Creamos unha para o estado do pulsador. Inicialmente apagado.
- ALUMNADO** Non podemos usar float para definirlo?
- PROFESORA** Podemos, pero neste caso non é necesario. Imos usar unha variable que só toma valores enteiros, así que non fai falta unha que nos dea decimais como é float.
- ALUMNADO** Ok
- PROFESORA** Agora imos aos bucles. Temos que definilos dentro de **void loop()**. Neste caso necesitamos un bucle **IF**. É un bucle que, se se cumpre a condición X, fará unha cousa, senón fará outra. Se o valor do pulsador é cero, acéndese a saída 3 (led verde) e as outras están apagadas.
- ALUMNADO** E cando non é cero?
- PROFESORA** Entón pasamos á parte **ELSE**. Nela apágase o led verde e acéndese un led rojo. Actívase un contador de tempo e cando pasa, invírtense os leds vermellos. Así, ata que cambie de novo a condición do bucle.

- ALUMNADO** Pero en que momento pasa iso?
- PROFESORA** Lembrades a variable que creamos co integer e que valía cero? Pois imos facer que en realidade sexa unha variable que depende da entrada do pin 2, así a modificamos co pulsador...
- ALUMNADO** Que chulo... Levounos algo de tempo, pero non foi tan difícil.

5. Conceptos básicos

Estructura básica dos programas de Arduino (void setup e loop setup)

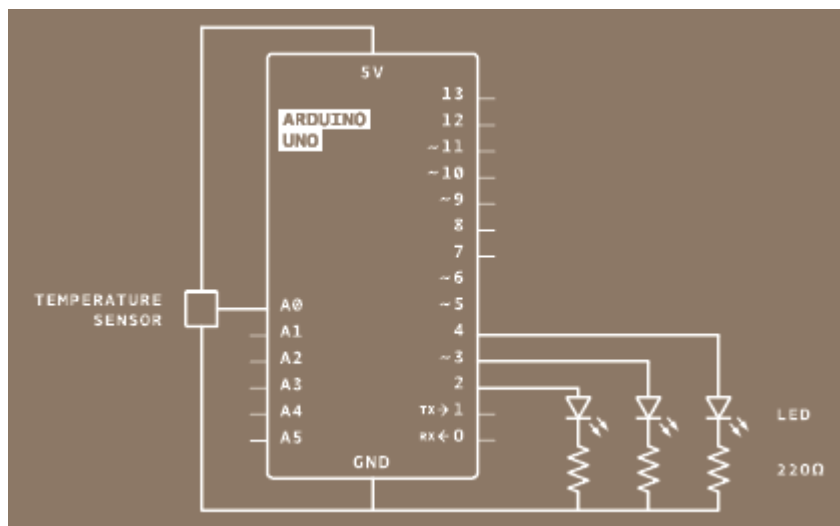
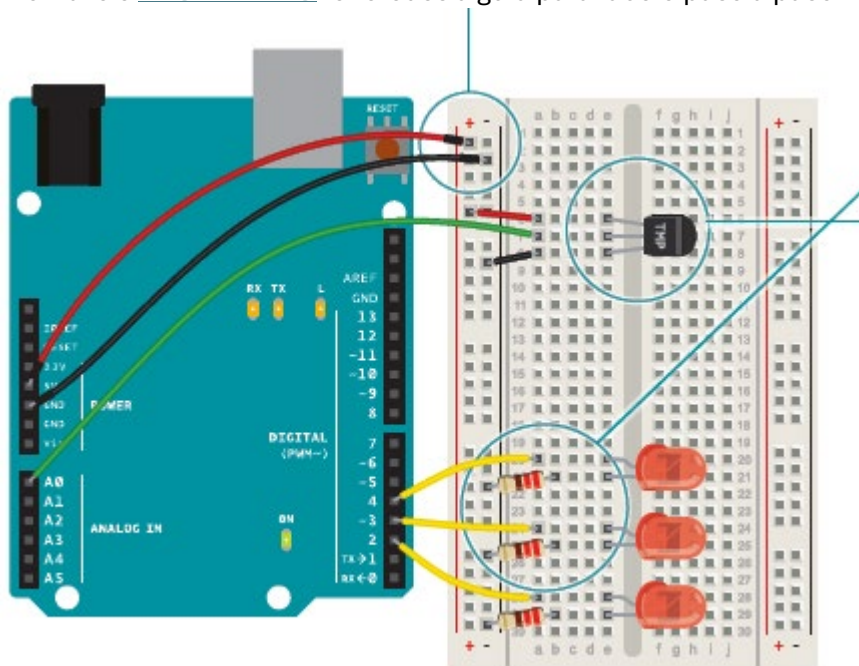
Declarar/crear variables

Configurar pines como entradas ou saídas

Bucles básicos (bucle if)

TEMA 6: AS ENTRADAS ANALÓXICAS

PROFESORA Bos días chic@s! Hoxe imos facer un proxecto no que aprenderemos a ler entradas analóxicas. O procedemento é o mesmo que a pasada semana. Déixovos por aquí o circuito que queremos crear, un esquema do montaxe e [NESTE VÍDEO](#) tenétedes a guía para facelo paso a paso.



Imos usar un medidor de temperatura e faremos que os leds se acendan en función da sinal. Canto maior sexa a temperatura, máis leds estarán acendidos.

Aquí tedes a folia de características do [SENSOR DE TEMPERATURA](#).

ALUMNADO Traballaremos pola nosa conta?

PROFESORA Por suposto. Xa podedes facelo. Ao finalizar repasaremos as cousas máis importantes deste proxecto.

...Alumnado traballando duramente... 😊

ALUMNADO Xa o rematamos e está a funcionar correctamente...

- PROFESORA** Perfecto. Os que non o teñan completado poden abrir o sketch deste exemplo en:
Arquivo – exemplos – StarterKit_BaseKit – p03_LoveOMeter
Pois fixédevos en varias cousas. Primeiro, para este proxecto, como estamos adquirindo unha sinal de maneira continua, é importante coñecer eses valores. Iso facémolo co **monitor serie** (ícono lupa arriba á dereita na ventana xeral). Iso abre unha ventana auxiliar para amostrar a sinal. Para iso necesitamos abrir o porto serie con **Serial.begin()**. O valor entre parénteses é a velocidade á que tomamos as mostras.
- ALUMNADO** Ok...
- PROFESORA** O sensor de temperatura fornecerá unha sinal analóxica proporcional á temperatura. Tal e como indica a folia de características. Debemos coñecer eses valores para poder calibrar o noso sensor.
- ALUMNADO** Fixádevos que creamos unha variable **sensorVal** co valor do sensor, que á súa vez é lido da entrada analóxica A0 mediante a función **analogRead**.
- PROFESORA** Con cada sensor analóxico necesitamos coñecer as súas curvas de calibración. Partindo da folia de características do sensor, temos que ser capaces de converter a súa sinal en voltaxe e posteriormente en algunha magnitude física. No noso caso, faise isto:
$$\text{float voltage} = (\text{sensorVal} / 1024.0) * 5.0$$
$$\text{float temperature} = (\text{voltage} - .5) * 100$$
- ALUMNADO** E se cambiamos a curva, cambiarían os valores.
- PROFESORA** Exacto. Cambiaremos segundo sexa necesario. Ás veces podemos axudarnos de Excel ou dalgún outro programa para crear as nosas curvas a partir dos datos do fabricante.
- ALUMNADO** Uf Excel... que medo...
- PROFESORA** Por último, neste programa volvemos a usar o bucle if, pero en lugar de só un, enlazamos varios empregando else if. Se queremos aplicar moitas condicións diferentes, é posible concatenar (unir) tantos bucles como sexan necesarios para cumprir o noso propósito.
- ALUMNADO** Sen límite de bucles?
- PROFESORA** Si, sen límite. Ademais, empregamos condicionales que debían cumprir máis dunha condición á vez, para o que engadimos estruturas lógicas **AND (&&)**. Con elas podemos pedir que se cumpran ao mesmo tempo 2 ou máis situacións. Tamén poderían ser **OR (||)**.
- ALUMNADO**

6. Conceptos básicos

Repasar a estrutura básica dos programas de Arduino

Monitor da porta serie

Declarar variables enteiras (int) e decimais (float)

Comando para ler entradas analóxicas

Curvas de calibración

Uso de bucles IF concatenados e condicións lóxicas (AND/OR)

TEMA 7: AXUDA COMPLEMENTARIA

7. Saídas analóxicas

[Vídeo Guía Arduino](#)

[Info Luis Llamas](#)

8. Visualización de datos por pantalla

[Vídeo guía](#)

[Info Luis Llamas](#)

9. Almacenamento de Datos en microSD

[Vídeo guía](#)

[Info Luis Llamas](#)