INICIACIÓN A ARDUINO

**Propósito**

Conocer el funcionamiento de las cosas es algo que el ser humano se ha planteado desde el inicio de los tiempos; no hay más que mirar a nuestro alrededor donde abunda la automatización, la domótica, la interacción de las personas con las máquinas, la electrónica, la mecánica o la programación.

Casi cualquier proceso que podamos imaginar es dependiente de estas máquinas, por ejemplo; que suene el despertador a las 7am para que puedas ir al instituto. Ese reloj trabajó durante toda la noche para que pudiera avisarte a qué hora tenías que despertarte.

El propósito de estas fichas consiste en abordar el concepto de computación física, que es la capacidad de comunicación de una máquina con los humanos, usando sensores y actuadores. Las decisiones que pretendamos transmitir, serán procesadas por un microcontrolador que se encuentra ubicado en la placa Arduino.

**¿Qué voy a aprender?**

Queremos que con el desarrollo de estas fichas con ejemplos prácticos entiendas que muchos procesos tecnológicos son simples de entender y no requieren de gran habilidad técnica y de un gran conocimiento, ya que para aquellos que son más complejos son la unión de muchos de los procesos simples.

La idea con estas fichas es que aprendas a imaginar y puedas convertir todas esas ideas que puedas tener a conceptos tangibles para que puedas verlo y tocarlo y entender el por qué de su funcionamiento.

**FICHA 1: Introducción a Arduino. Programación básica**

1. **¿Qué es Arduino?**

Arduino es una plataforma open-hardware basada en una sencilla placa con ***entradas*** y **salidas** (E/S), analógicas y digitales, y en un entorno de desarrollo que implementa el lenguaje Processing. Al ser open-hardware tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para desarrollar cualquier tipo de proyecto sin tener que adquirir ningún tipo de licencia. Asimismo, su sencillez y su bajo coste, recomiendan su uso como elemento de aprendizaje e iniciación en el mundo de la electrónica digital.



1. **Conociendo el Software Arduino**

**Pasos para arrancar nuestro Arduino**

1. Para ejecutar el programa Arduino, ingresaremos en la carpeta Arduino y allí buscaremos el icono: 
2. Una vez cargado nos abrirá una pantalla donde tendremos que escribir el código:

|  |  |
| --- | --- |
|  |  |
|  |
|  |
| Guardar |
| Abrir |
| Nuevo |
| **Cargar Placa** |
| **Compilar** |
|  |
|  |

1. Define el puerto USB al que está conectado tu placa Arduino. Tools > Serial Port > COM (el que detecte)



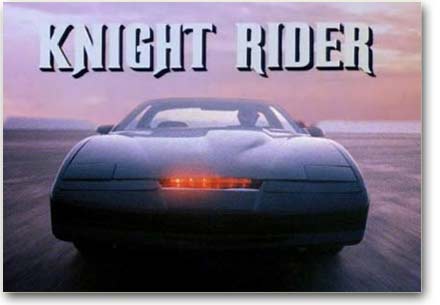
1. Selecciona la placa Arduino que vas a trabajar, en nuestro caso será la Arduino Uno. Tools > Board > Arduino Uno



1. Cargar el programa. Una vez hemos configurado nuestras especificaciones de la placa y hemos escrito el código que queremos que el programa nos ejecute, lo cargaremos a la placa.



Una vez que la carga a la placa concluya, aparece un mensaje Carga Completada. En su defecto, si el programa no fuera correcto se te informará del error en la zona de mensajes del software.

1. **Programación básica.**

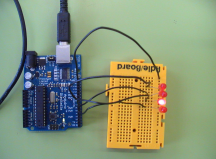
En esta primera sesión vamos a retroceder a los años 80 para aprender más acerca de la programación secuencial y buenas técnicas de programación para la placa E/S, con un ejemplo interesante como el del ***Coche Fantástico****.* El coche tenía en su parte delantera una multitud de LEDs de todos los tamaños realizando efectos de luces.

En la simulación utilizaremos 4 LEDs conectados a los pins de la placa ShieldBoard, los cuales ya tienen integradas resistencias de 220 Ohm.



El primer código del ejemplo, hará que todas los LEDs se enciendan y se apaguen por un tiempo determinado, usando *digitalWrite (pinNum, HIGH/LOW)* y *delay(time).* En el segundo código, se desarrollará una secuencia de encendido LED por LED y posteriormente del mismo modo se apagará. En el tercer y último ejemplo se introducirá un pulsador que hará que una luz que está parpadeando se encienda por completo y la inmediatamente después empiece a parpadear, y así hasta completar la secuencia, volviendo después a su estado inicial.

* **Objetivo de la sesión:** ejemplos de un juego de secuencia con diodos LED para la que se utilizará la Shield Board como introducción a las posibilidades que ofrece Arduino sin empezar a cablear sobre la Breadboard.
* **Objetivo didáctico:**  En esta primera fase de aprendizaje de Arduino el alumno tiene que llegar a entender el código que se carga en la placa Arduino y descubrir el funcionamiento o secuencia que llevarán a cabo los diodos LED.
* **Componentes necesarios:**
* Placa Arduino Uno.
* Shield Board
* Clabe USB
* Pulsadores

***Ejemplo 1***

Se trata que el Coche Fantástico encienda y apague 4 leds simultaneamente. Los leds tienen asignada la conexión en la Shieldboard a los pines 11,10,9 y 6. Se deben encender y posteriormente apagar los LEDs con un tiempo de duración de encendido y apagado de 1000 milisegundos.

|  |  |
| --- | --- |
| Plano de conexiones | esquema electrico_circuit_1y2.jpg |
|  |  |
| Esquema del montaje sobre Breadboard | esquema_montaje.jpg |

|  |  |
| --- | --- |
| CÓDIGO DE PROGRAMACIÓN | // Juego de Luces 1 – Coche Fantastico  // Esto son lineas comentadas, las cuales empiezan con "//" y no afectan al programa.  // Ahora empecemos a analizar el programa. Primero vamos a empezar con asignaciones (variables globales) y  //ver qué es lo que tenemos.  // Estas lineas son las asignaciones de los LED's Las ponemos como constantes "const" para que no puedan  //cambiar.    **const int** ROJO = 11; // Asignamos al [COLOR] el valor [x], que tiene relación con el pin en el que se encuentra  **const int** AMBAR = 10;  **const int** VERDE = 9;  **const int** BLANCO = 6;  // Ahora vamos con la parte básica del programa. Una función es un bloque que hace una tarea en concreto.  // ***setup()*** es una función que se ejecuta una sola vez cuando el programa comienza. Se utiliza para cosas que hay  //que hacer al principio y solo una vez.    void **setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode es una función que define un pin como entrada o salida. Ponemos  **pinMode**(BLANCO, **OUTPUT**); // pinMode (el pin sobre el que queremos [actuar], INPUT (entrada) / OUTPUT  **pinMode**(VERDE, **OUTPUT**); //(salida)  **pinMode**(BLANCO, **OUTPUT**);  }    // ***loop()*** es una función que se ejecuta permanentemente hasta que desconectas arduino o pulsas reset  void **loop**()  {  **int** Tiempo = 1000; // Asignamos el tiempo que queremos que mantenga ese estado en milisegundos    **digitalWrite**(ROJO, **HIGH**); //digitalWrite lee el pinMode y circula corriente(HIGH)  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  **digitalWrite**(BLANCO, **HIGH**);  delay(Tiempo); // variable del tiempo de ejecución del estado  **digitalWrite**(ROJO, **LOW)**; // deja de circular corriente por ese camino(LOW)  **digitalWrite**(AMBAR, **LOW**);  **digitalWrite**(VERDE, **LOW**);  **digitalWrite**(BLANCO, **LOW**);  delay(Tiempo);  } |

***Ejemplo 2***

Mantiene la misma forma del ejemplo anterior pero en este caso se trata de encender y apagar 4 leds secuencialmente.

|  |  |
| --- | --- |
| CÓDIGO DE PROGRAMACIÓN | // Juego de Luces 2 – Coche Fantastico – encendido/apagado secuencial  //Asignacion de los LED's como constantes "const" para que no puedan cambiar.  **const int** ROJO = 11; //Determina donde estan conectados los pines  **const int** AMBAR = 10;  **const int** VERDE = 9;  **const int** BLANCO = 6;  // ***setup()*** es una función que se ejecuta una sola vez cuando el programa comienza.  **void setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode es una función que define un pin como entrada o salida. Ponemos  **pinMode**(BLANCO, **OUTPUT**); // pinMode( [el pin sobre el que queremos actuar], INPUT (entrada) / OUTPUT  **pinMode**(VERDE, **OUTPUT**); //(salida)  **pinMode**(BLANCO, **OUTPUT**);  }    **void loop**()  {  **int** Tiempo = 250; // Asignar el tiempo que determina la acción de la siguiente variable    **digitalWrite**(ROJO, **HIGH**); //digitalWrite lee el pinMode y circula corriente(HIGH  delay(Tiempo); // variable del tiempo de ejecución hasta pasar al siguiente  **digitalWrite**(AMBAR, **HIGH**);  delay(Tiempo);  **digitalWrite**(VERDE, **HIGH**);  delay(Tiempo);  **digitalWrite**(BLANCO, **HIGH**);  delay(Tiempo);  **digitalWrite**(BLANCO, **LOW**); // deja de circular corriente por ese camino(LOW) por ese pin  delay(Tiempo);  **digitalWrite**(VERDE, **LOW**);  delay(Tiempo);  **digitalWrite**(AMBAR, **LOW**);  delay(Tiempo);  **digitalWrite**(ROJO, **LOW**);  delay(Tiempo);    } |

***Ejemplo 3***

En este ejemplo introduciremos una nueva variable de entrada (INPUT) que será el encargado de activar/desactivar los LEDs. Con el fin que nuestro coche se mantenga siempre en estado de alerta, dejaremos que mientras el pulsador no se accione un LED siempre se mantenga en estado de parpadeo. Una vez se pulse el botón, se encenderá por completo mientras el que está a su lado se activará en espera de accionar nuevamente el pulsador y así sucesivamente.

|  |  |
| --- | --- |
| Plano de conexiones | esquema electrico_circuit_3.jpg |
|  |  |
| Esquema del montaje sobre Breadboard | esquema_montaje_circuit2.jpg |
| CÓDIGO DE PROGRAMACIÓN | // Juego de luces 3 – Coche Fantastico – parpadeo de una luz hasta que el pulsador la deje fija y  // comience el parpadeo de la inmediatamente después.  //Asignacion de los LED's como constantes "const" para que no puedan cambiar.  **const int** ROJO = 11;  **const int** AMBAR = 10;  **const** int VERDE = 9;  **const** int BLANCO = 6;  **const** int PULSADOR = 12;      **void setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode es una función que define un pin como entrada o salida.  **pinMode**(BLANCO, **OUTPUT**); //Ponemos pinMode( [el pin sobre el que queremos actuar], INPUT  **pinMode**(VERDE, **OUTPUT**); //(entrada) / OUTPUT (salida)  **pinMode**(BLANCO, **OUTPUT**);  **pinMode**(PULSADOR, **INPUT**);  }    **void loop()**    {  **int** Pulso;  **int** Veces=0; // Variables que vamos a utilizar.  **int** Vuelta = 300; // Tiempo de parpadeo    **digitalWrite**(ROJO, **LOW**); // Primero dejamos todos los LED's apagados  **digitalWrite**(AMBAR, **LOW**);  **digitalWrite**(VERDE, **LOW**);  **digitalWrite**(BLANCO, **LOW**);  delay(10);        while(Veces < 5) // El ciclo ***while*** funciona de tal manera que al hacer una acción mientras la  //condición (este caso que "Veces" sea menor a 5) sea cierta. La variable  //"Veces" recoje las veces que se ha pulsado un pulsador.  {  Pulso = **digitalRead**(PULSADOR); // Lo mismo de antes, solo que creamos la variable "Pulso" para  //guardar el valor leido del pulsador 2.  if (Pulso == **LOW**) // Si pulsamos el pulsador 2...  {Veces = Veces + 1;} // A "Veces" le sumamos 1. La primera vez, como empieza de 0, hace  // "Veces = 0 + 1", la segunda vuelta, "Veces" será igual a 1 por lo tanto,  // hará "Veces = 1 + 1", y tomará el valor 2.  // Pero... ¿cuándo acaba el bucle "while"? Esto se da cuando la condición que tiene (que "Veces"  //sea menor a 5) no se cumpla (por ejemplo en "Veces" = 5, 6...). Lo que hace es inmediatamente  //salir de lo que tenga dentro (el "switch" con sus case).    // "switch" es muy parecido a "if", solo que contempla más casos. Depende de la variable "Veces".  //Esto significa que cuando "Veces" sea 0, el programa hará lo que ponga en "case 0:" y saldrá,  //cuando "Veces" sea 1, hará lo que mande case 1: y saldrá... Hasta que la condición del "while" sea  //falsa.        **switch**(Veces)  {  case 0:  delay(Vuelta);  break;    case 1:  **digitalWrite**(ROJO, **HIGH**); // Encender y apagar el LED Rojo con un retardo.  delay(Vuelta);  **digitalWrite**(ROJO, **LOW**);  delay(Vuelta);  break;    case 2:  **digitalWrite**(ROJO, **HIGH**); // Mantener encendido el LED Rojo y hacer parpadear el LED Ambar.  **digitalWrite**(AMBAR, **HIGH**);  delay(Vuelta);  **digitalWrite(**AMBAR, **LOW**);  delay(Vuelta);  break;    case 3:  **digitalWrite**(ROJO, **HIGH**); // Mantener el Rojo y Ambar y parpadear el Verde.  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  delay(Vuelta);  **digitalWrite**(VERDE, **LOW**);  delay(Vuelta);  break;    case 4:  **digitalWrite**(ROJO, **HIGH**); // Mantener, Rojo, Ambar y Verde y parpadear el Blanco.  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  **digitalWrite**(BLANCO, **HIGH**);  delay(Vuelta);  **digitalWrite**(BLANCO, **LOW**);  delay(Vuelta);  break;  }  }  delay(Vuelta); // Reposo para que no salte inmediatamente.  } |

***Ejercicio para proponer***

El objetivo de este ejercicio propuesto es crear, teniendo en cuenta los conceptos adquiridos, la secuencia famosa del Coche Fantástico de hacer que los LED parpadeen en una secuencia, uno por uno e ida y vuelta usando sólo **digitalWrite(pinNum,HIGH/LOW)** y **delay(time)**. El esquema del montaje será el mismo que en los ejemplos 1 y 2.

Hay que tener en cuenta la siguiente asignación de los LEDs:

**const int ROJO = 11;**

**const int AMBAR = 10;**

**const int VERDE = 9;**

**const int BLANCO = 6;**

**int Timer = 250;**

Resultado:

|  |  |
| --- | --- |
| CÓDIGO DE PROGRAMACIÓN | const int ROJO = 11;  const int AMBAR = 10;  const int VERDE = 9;  const int BLANCO = 6;  int timer = 250;  void setup(){  pinMode(ROJO, OUTPUT);  pinMode(AMBAR, OUTPUT);  pinMode(VERDE, OUTPUT);  pinMode(BLANCO, OUTPUT);  }  void loop() {  digitalWrite(ROJO, HIGH);  delay(timer);  digitalWrite(ROJO, LOW);  delay(timer);  digitalWrite(AMBAR, HIGH);  delay(timer);  digitalWrite(AMBAR, LOW);  delay(timer);  digitalWrite(VERDE, HIGH);  delay(timer);  digitalWrite(VERDE, LOW);  delay(timer);  digitalWrite(BLANCO, HIGH);  delay(timer);  digitalWrite(BLANCO, LOW);  delay(timer);  digitalWrite(VERDE, HIGH);  delay(timer);  digitalWrite(VERDE, LOW);  delay(timer);  digitalWrite(AMBAR, HIGH);  delay(timer);  digitalWrite(AMBAR, LOW);  delay(timer);  } |

ARDUINO HASTAPENAK

**Helburua**

Gauzak nola lan egiten duten jakitea existentzia hasieratik gizakiak erakarri duen zerbait da; automozioa, domotika, makina eta pertsona arteko elkarrekintza, elektronika, mekanika edo programazioari besterik ez dugu erreparatu behar.

Imajinatu dezakegun edozein gauza makina menpean dago; adibidez, institutura joateko iratzargailua goizeko 7tan jotzeak. Erloju hori gau osoan zehar lanean haritu da esnatu behar duzun orduan abisatu ahal izateko.

Fitxa hauen helburua informatika fisikoaren kontzeptuari aurre egitean datza, eta hori gizaki eta makinen arteko komunikazio gaitasuna da, sentsore eta eragingailuak erabiliz. Igorri nahi ditugun erabakiak, Arduino plakak daukan mikrokontrolagailuak prozesatuko ditu.

**Zer ikasiko dut?**

Adibide praktiko batzuen bitartez, prozesu teknologiko asko ulertzeko sinpleak direnak eta maila tekniko eta ezagutza handia eskatzen ez dituztenak, konplexuagoak diren prozesu askoren batasunak besterik ez direla ulertarazi nahi dugu.

Fitxa hauen helburua imajinatzen duzun edozein prozesu, kontzeptu ukigai bihurtzeko gaitasuna lortzean datza eta zergati guztiak ulertzeko aukera eskainiko dizkizu ikasten duzun bitartean.

**FITXA 1: Arduino Hastapenak. Oinarrizko proramazioa.**

1. **Arduino zer da?**

Arduino open-hardware platarforma da, sarrera eta irteera (E/S), analogiko eta digitaletan, oinarrituta dagoen plaka da, Processing hizkuntza garapenerako ingurune batean erabilita. Open-hardware bai diseinuan zein fabrikazioa doakoa da. Hau da, edozein lizentzia erosi gabe libreki edozein proiektu garatzeko erabili daiteke. Gainera, bere sinpletasun eta kostu txikia, bere ikaskuntza eta hastapen elektronika digital mundura elementu gisa erabiltzeko gomendagarria bihurtzen da.



1. Arduino Softwarerra ezagutzen

Arduinoa abiarazteko jarraitu beharreko pausoak

1. Arduino programa irekitzeko, Arduino karpetan sartuko gara eta bertan  ikonoa bilatuko dugu, bertan klikatuz
2. Behin ireki dugunean, kodea idatziko dugun pantaila agertuko da:

|  |  |
| --- | --- |
|  |  |
|  |
|  |
| Gorde |
| Ireki |
| Berria |
| **Plakan kargatu** |
| **Compilatu** |
|  |
|  |

1. Arduino plaka konektatuta dagoen USB tokia egokitu. Tools > Serial Port > COM (aukeratu eskaintzen duen aukera)



1. Aukeratu lan egingo dugun Arduino plaka, gure kasuan Arduino Uno izango da.

Tools > Board > Arduino Uno



1. Programa kargatu. Behin gure plakaren zehaztapenak konfiguratu ditugula eta kodea idatzita, plakari kargatuko diogu.



Karga bukatzerakoan, Karga burutu den mezua agertuko da. Horrela ez bada, kodeak akatsaren bat duela abisatuko dizu programak berak eta era berean konponbidea bilatzen lagunduko dizu ohar bidez.

1. Oinarrizko programazioa

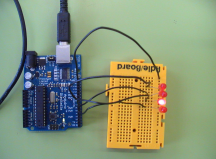
Lehenengo saio honeta, 80ko hamarkadara itzuliko gara programazio sekuentziala eta E/S plakaren programazio teknika ikasteko, Kotxe Fantastikoa adibide interesgarri gisa erabiliz. Autoak bere aurreko muturrean, argi efektu joko karakteristikoa egiten zituen tamaina guztiko LED argiak zituen.

Gure simulazioan ShieldBoard plakan konektatuta dauden 4 LED bere (220Ohm erresistentziak berne dituztela) erabiliko ditugu gure sekuentziekin jolasteko.



Erakutsiko dugun lehenengo kodean, LED guztiak aldi berean piztu eta itzaliko dira definituko diogun denbora pasa eta gero, *digitalWrite (pinNum, HIGH/LOW)* eta *delay(time)* erabiliz. Bigarren adibidean, LEDz LED piztu eta itzaltzen den sekuentzia egingo dugu. Azken adibidean, pultsadore bat sartuko dugu, non argi bat kliskatzen egongo da pultsadoreari eman arte. Behin sakatuta, argi horrek finko geratuko da eta ondorengoa kliskatzen hasiko da.

* **Saioaren helburua:** LED argiaz osatutako adibide jokoak egingo dira. Horretarako, ShieldBoard erabiliko da sarrera gisa, Arduinok aurkezten dituen aukeren artean, kableekin zirkuituak eraikitzen hasi aurretik.
* **Helburu didaktikoa:** Arduino ikasketa lehengo saio honetan ikasleak kodeak Arduino plakan nola kargatu egiten den ikasi beharko du eta LED argiak jarraituko duen sekuentziaz jabetu segun kodeak jasotzen duen moduan.
* **Beharrezko materiala:**
* Arduino Uno plaka
* ShieldBoard
* USB kablea
* Pultsadorea

Adibidea 1 bilatu

Gure kotxe fantastikoak aldi berean 4 LED argi piztean datza aldi berean. LED argiak ShieldBoard-en 11,10,9 eta 6 pinak egokituta dauzka. Argi guztiak 1000 milisegundu (segundo bat) denbora pasa ostean piztu-itzali sekuentzia burutu beharko du.

|  |  |
| --- | --- |
| Konexiorako planoa | esquema electrico_circuit_1y2.jpg |
|  |  |
| Breadboard gaineko eskema | esquema_montaje.jpg |

|  |  |
| --- | --- |
| PROGRAMAZIO KODEA | // Argi jokoa 1 – Kotxe fantastikoa  // "//" komentateko lerroak dira, programa kaltetzen ez dituztenak  // Has gaitezen programa aztertzen. Lehendabizi, aldagaiak (aldagai globalak) definituko ditugu. Aldagai hauek  // LED argiak izango dira eta algaezin egingo ditugu “const” erabiliz.    **const int** ROJO = 11; // [KOLOREA] [x]-ri egokitzen diogu, kokatuta dagoen pinarekin harremana duena  **const int** AMBAR = 10;  **const int** VERDE = 9;  **const int** BLANCO = 6;  // Programaren zati oinarrizkoena. Funtzioa bloke batek lan bat egiten duena da.  //***setup()*** programa hasieran behin bakarrik erabiltzen den funtzioa da. Gero ez da berriz erabiltzen    void **setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode pin bat sarrera edo irteera moduan definitzen duen funtzioa da.  **pinMode**(BLANCO, **OUTPUT**); // pinMode(zein pinaren gainean aritu nahi dugun, INPUT (sarrera)/OUTPUT  **pinMode**(VERDE, **OUTPUT**); //(irteera)  **pinMode**(BLANCO, **OUTPUT**);  }    // ***loop()*** etengabe aritzen den funtzioa da, plaka deskonektatu arte edo reset eman arte  void **loop**()  {  **int** Tiempo = 1000; // Zenbat denboraz aritu nahi den hurrengo etapara pasa arte definitzen da.    **digitalWrite**(ROJO, **HIGH**); //digitalWriten, pinMode-a irakurri eta korrontea igarotzen da (HIGH)  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  **digitalWrite**(BLANCO, **HIGH**);  delay(Tiempo); // denbora aldagaia gauzatze egoera.  **digitalWrite**(ROJO, **LOW)**; // korrontea moztu egiten (LOW) bide horretan  **digitalWrite**(AMBAR, **LOW**);  **digitalWrite**(VERDE, **LOW**);  **digitalWrite**(BLANCO, **LOW**);  delay(Tiempo);  } |

***Adibidea 2***

Aurreko egitura berdina mantenduz, sekuentzia bat jarraituz piztu eta itzali 4 LED argiak.

|  |  |
| --- | --- |
| PROGRAMAZIO KODEA | // Argi jokoa 2 – Kotxe fantastikoa – piztu/itzali sekuentziala  // LED argiak izango dira eta algaezin egingo ditugu “const” erabiliz.  **const int** ROJO = 11; //pin-ak non dauden konektatuta zehazten du.  **const int** AMBAR = 10;  **const int** VERDE = 9;  **const int** BLANCO = 6;  // ***setup()*** hasieran bakarrik gauzatzen den funtzioa da.  **void setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode pin bat sarrera edo irteera moduan definitzen duen funtzioa da.  **pinMode**(BLANCO, **OUTPUT**); // pinMode(zein pinaren gainean aritu nahi dugun, INPUT (sarrera)/OUTPUT  **pinMode**(VERDE, **OUTPUT**); //(irteera)  **pinMode**(BLANCO, **OUTPUT**);  }    **void loop**()  {  **int** Tiempo = 250; // Hurrengo aldagai ekintza zehazten duen denbora da.    **digitalWrite**(ROJO, **HIGH**); //digitalWriten, pinMode-a irakurri eta korrontea igarotzen da (HIGH)  delay(Tiempo); // denbora algaia hurrengo egoerara pasa arte.  **digitalWrite**(AMBAR, **HIGH**);  delay(Tiempo);  **digitalWrite**(VERDE, **HIGH**);  delay(Tiempo);  **digitalWrite**(BLANCO, **HIGH**);  delay(Tiempo);  **digitalWrite**(BLANCO, **LOW**); // korrontea moztu egiten (LOW) bide horretan  delay(Tiempo);  **digitalWrite**(VERDE, **LOW**);  delay(Tiempo);  **digitalWrite**(AMBAR, **LOW**);  delay(Tiempo);  **digitalWrite**(ROJO, **LOW**);  delay(Tiempo);    } |

***Adibidea 3***

Adibide honetan, sarrera aldagai berri bat sartuko dugu (INPUT) LED argiak aktibatzen gaitasuna izango duena. Gure kotxeak alerta-egoera batean egon dadin, LED argi bat beti kliskatuta egongo da pultsadorea sakatu ez den bitartean. Behin etengailua sakatuta, kliska egiten zegoen argia guztiz piztuko da bere ondorengoa kliska egoera jartzen den bitartean, berriz ere pultsadorea sakatu arte.

|  |  |
| --- | --- |
| Konexiorako planoa | esquema electrico_circuit_3.jpg |
|  |  |
| Breadboard gaineko eskema | esquema_montaje_circuit2.jpg |
| CÓDIGO DE PROGRAMACIÓN | // Argi jokoa 3 - Kotxe fantastikoa – argia kliskatu egoera batean mantendu pultsarea sakatu arte, fijo  // gelditu arte, ondorengoa kliskatu egoera hasten den einean.  // LED argiak izango dira eta algaezin egingo ditugu “const” erabiliz.  **const int** ROJO = 11;  **const int** AMBAR = 10;  **const** int VERDE = 9;  **const** int BLANCO = 6;  **const** int PULSADOR = 12;      **void setup**()  {  **pinMode**(ROJO, **OUTPUT**); // pinMode pin bat sarrera edo irteera moduan definitzen duen funtzioa da.  **pinMode**(BLANCO, **OUTPUT**); // pinMode(zein pinaren gainean aritu nahi dugun, INPUT (sarrera)/OUTPUT  **pinMode**(VERDE, **OUTPUT**); // (irteera)  **pinMode**(BLANCO, **OUTPUT**);  **pinMode**(PULSADOR, **INPUT**);  }    **void loop()**    {  **int** Pulso;  **int** Veces=0; // Erabiliko ditugun aldagaiak  **int** Vuelta = 300; // kliskatu denbora definitu    **digitalWrite**(ROJO, **LOW**); // Lehendabizi LED argi guztiak itzalita utziko ditugu.  **digitalWrite**(AMBAR, **LOW**);  **digitalWrite**(VERDE, **LOW**);  **digitalWrite**(BLANCO, **LOW**);  delay(10);        while(Veces < 5) // ***while*** funtzioa Veces <5 aginpean burutu egingo den sekuentzia da.  //”Veces” pultsadoreak zenbat aldiz sakatu den hartzen du  {  Pulso = **digitalRead**(PULSADOR); // Aurrekoaren verdina, baina “Pulso” aldagaia sortu da non  //guardar el valor leido del pulsador 2.  if (Pulso == **LOW**) // Si pulsamos el pulsador 2...  {Veces = Veces + 1;} // A "Veces" le sumamos 1. La primera vez, como empieza de 0, hace  // "Veces = 0 + 1", la segunda vuelta, "Veces" será igual a 1 por lo tanto,  // hará "Veces = 1 + 1", y tomará el valor 2.  // Pero... ¿cuándo acaba el bucle "while"? Esto se da cuando la condición que tiene (que "Veces"  //sea menor a 5) no se cumpla (por ejemplo en "Veces" = 5, 6...). Lo que hace es inmediatamente  //salir de lo que tenga dentro (el "switch" con sus case).    // "switch" es muy parecido a "if", solo que contempla más casos. Depende de la variable "Veces".  //Esto significa que cuando "Veces" sea 0, el programa hará lo que ponga en "case 0:" y saldrá,  //cuando "Veces" sea 1, hará lo que mande case 1: y saldrá... Hasta que la condición del "while" sea  //falsa.        **switch**(Veces)  {  case 0:  delay(Vuelta);  break;    case 1:  **digitalWrite**(ROJO, **HIGH**); // Encender y apagar el LED Rojo con un retardo.  delay(Vuelta);  **digitalWrite**(ROJO, **LOW**);  delay(Vuelta);  break;    case 2:  **digitalWrite**(ROJO, **HIGH**); // Mantener encendido el LED Rojo y hacer parpadear el LED Ambar.  **digitalWrite**(AMBAR, **HIGH**);  delay(Vuelta);  **digitalWrite(**AMBAR, **LOW**);  delay(Vuelta);  break;    case 3:  **digitalWrite**(ROJO, **HIGH**); // Mantener el Rojo y Ambar y parpadear el Verde.  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  delay(Vuelta);  **digitalWrite**(VERDE, **LOW**);  delay(Vuelta);  break;    case 4:  **digitalWrite**(ROJO, **HIGH**); // Mantener, Rojo, Ambar y Verde y parpadear el Blanco.  **digitalWrite**(AMBAR, **HIGH**);  **digitalWrite**(VERDE, **HIGH**);  **digitalWrite**(BLANCO, **HIGH**);  delay(Vuelta);  **digitalWrite**(BLANCO, **LOW**);  delay(Vuelta);  break;  }  }  delay(Vuelta); // Reposo para que no salte inmediatamente.  } |

***Proposatuko ariketa***

Ariketa honen helburua, ikasi diren kontzeptuekin, kotxe fantastikoaren joan-etorri LED argiaren sekuentzia eraikitzea,  **digitalWrite(pinNum,HIGH/LOW)**  eta **delay(time) funtzioak erabiliz. Muntaia eta eskema adibide 1 eta 2 ikusi dugunaren berdina**  izango da eta hurrengo aldagaiak kontuan izango dira:

**const int ROJO = 11;**

**const int AMBAR = 10;**

**const int VERDE = 9;**

**const int BLANCO = 6;**

**int Timer = 250;**

Emaitza:

|  |  |
| --- | --- |
| PROGRAMAZIO KODEA | const int ROJO = 11;  const int AMBAR = 10;  const int VERDE = 9;  const int BLANCO = 6;  int timer = 250;  void setup(){  pinMode(ROJO, OUTPUT);  pinMode(AMBAR, OUTPUT);  pinMode(VERDE, OUTPUT);  pinMode(BLANCO, OUTPUT);  }  void loop() {  digitalWrite(ROJO, HIGH);  delay(timer);  digitalWrite(ROJO, LOW);  delay(timer);  digitalWrite(AMBAR, HIGH);  delay(timer);  digitalWrite(AMBAR, LOW);  delay(timer);  digitalWrite(VERDE, HIGH);  delay(timer);  digitalWrite(VERDE, LOW);  delay(timer);  digitalWrite(BLANCO, HIGH);  delay(timer);  digitalWrite(BLANCO, LOW);  delay(timer);  digitalWrite(VERDE, HIGH);  delay(timer);  digitalWrite(VERDE, LOW);  delay(timer);  digitalWrite(AMBAR, HIGH);  delay(timer);  digitalWrite(AMBAR, LOW);  delay(timer);  } |