

17-5-2021

Lenguaje de programación LYADRONE

Lenguajes y Autómatas I



Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

5A 2:00 pm

Docente: M.D.O.H Sonia Alvarado Mares

Índice

Introducción	3
Filosofía	3
Alfabeto	3
Palabras reservadas	4
Tipos de datos	9
Cadenas	9
Números	9
Operadores	10
Identificadores	11
Variables	11
Arreglos	12
Funciones	12
Llamar una función	13
Fin de una sentencia	13
Comentarios	13
Condicionales	14
si	14
si no	14
Si y si no anidados	15
Ciclos	16
ciclo	16
mientras	16
haz - mientras	17
Constantes del lenguaje	17
apagar	18
encender	18
calibrar	18
direccionar	18
arriba	18

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

abajo.....	18
izquierda.....	18
derecha.....	18
adelante	18
atras.....	18
girar	19
fijo.....	19
acelerar.....	19
detener	19
esperar	19
seguir.....	19
pin.....	20
leer	20
imprimir.....	20
captura	20
video.....	20
parav.....	20
ruta	21
Formato del script	21
Cargar librerías	23
Bibliografía	25

Introducción

Con el propósito de desarrollar un compilador, se encomendó la tarea de iniciar el proyecto creando un lenguaje de programación que se pueda usar en microcontroladores. El que esté hecho de esta manera hace especial al lenguaje ya que como algunos de los propósitos es que sea lo más parecido al lenguaje natural o técnicamente hablando que sea de alto nivel, pero que al mismo tiempo pueda tener esa parte que manipule el hardware que se use dependiendo de la aplicación que se le vaya a dar.

De esta manera nació el lenguaje de programación **LYADROME**, y en este documento se dará una introducción a lo que es lenguaje, cómo usarlo, y de cómo se puede implementar en la creación y manipulación de drones aprovechando las características del lenguaje y funciones definidas que harán la programación más sencilla, dejando que el programador solo se enfoque en la lógica que tendrá que implementar y no en que herramientas usar.

Filosofía

LYADROME es un lenguaje diseñado en español, con una sintaxis muy intuitiva que les permitirá a los desarrolladores crear código para proyectos que involucren microcontroladores y su implementación en desarrollo en drones.

Para diseñar el lenguaje se realizó una breve investigación sobre la sintaxis de los lenguajes más usados, y se seleccionaron las principales características que se observaron hacen que los programadores opten por usarlos. Con base a estas ideas es que se crea **LYADROME**.

Alfabeto

- Letras permitidas: {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}

*nota: solo se permiten letras en minúscula

- Dígitos: {0,1,2,3,4,5,6,7,8,9}

Daniel Alejandro Calderón Virgen 18401090
Luis Sergio Carrillo Sánchez 18401095
Darien Rafael Márquez Vázquez 17401326
Diego Arturo Cadena Flores 18401089

- Operadores y símbolos especiales {(,)=,+,-,>,<.,*,,,,|,&^,@,%,',_}
- Separadores {\n (Nueva línea), \r (Retorno de carro), \t (Tabulador horizontal), (espacio en blanco)}

Palabras reservadas

<i>abajo</i>	<i>acelerar</i>	<i>adelante</i>	<i>apagar</i>	<i>arreglo</i>	<i>arriba</i>
<i>atras</i>	<i>booleano</i>	<i>cadena</i>	<i>calibrar</i>	<i>captura</i>	<i>captura</i>
<i>caracter</i>	<i>cargar</i>	<i>ciclo</i>	<i>corto</i>	<i>decimal</i>	<i>declaracion</i>
<i>derecha</i>	<i>detener</i>	<i>direccionar</i>	<i>ejecutar</i>	<i>encender</i>	<i>entero</i>
<i>esperar</i>	<i>fijo</i>	<i>fin</i>	<i>funcion</i>	<i>girar</i>	<i>haz</i>
<i>imprimir</i>	<i>iniciar</i>	<i>izquierda</i>	<i>largo</i>	<i>leer</i>	<i>mientras</i>
<i>no</i>	<i>nulo</i>	<i>parav</i>	<i>pin</i>	<i>principal</i>	<i>regresa</i>
<i>retorno</i>	<i>ruta</i>	<i>seguir</i>	<i>si</i>	<i>video</i>	<i>video</i>
<i>falso</i>	<i>verdadero</i>				

Patrones de la palabras reservadas

Palabra Reservada	Patrón
<i>atras</i>	Cadena que comienza con la letra a, seguida de la letra t, seguida de la letra r, seguida de la letra a y termina con la letra s
<i>adelante</i>	Cadena que comienza con la letra a, seguida de la letra d, seguida de la letra e, seguida de la letra l, seguida de la letra a, seguida de la letra n, seguida de la letra t y termina con la letra e
<i>apagar</i>	Cadena que comienza con la letra a, seguida de la letra p, seguida de la letra a, seguida de la letra g, seguida de la letra a y termina con la letra r
<i>arreglo</i>	Cadena que comienza con la letra a, seguida de la letra r, seguida de la letra r, seguida de la

	letra e, seguida de la letra g, seguida de la letra l y termina con la letra o
abajo	Cadena que comienza con la letra a, seguida de la letra b, seguida de la letra a, seguida de la letra j y termina con la letra o
acelerar	Cadena que comienza con la letra a, seguida de la letra c, seguida de la letra e, seguida de la letra l, seguida de la letra e, seguida de la letra r, seguida de la letra a y termina con la letra r
booleano	Cadena que comienza con la letra b, seguida de la letra o, seguida de la letra o, seguida de la letra l, seguida de la letra e, seguida de la letra a, seguida de la letra n y termina con la letra o
cadena	Cadena que comienza con la letra c, seguida de la letra a, seguida de la letra d, seguida de la letra e, seguida de la letra n y termina con la letra a
captura	Cadena que comienza con la letra c, seguida de la letra a, seguida de la letra p, seguida de la letra t, seguida de la letra u, seguida de la letra r y termina con la letra a
caracter	Cadena que comienza con la letra c, seguida de la letra a, seguida de la letra r, seguida de la letra a, seguida de la letra c, seguida de la letra t, seguida de la letra e y termina con la letra r
cargar	Cadena que comienza con la letra c, seguida de la letra a, seguida de la letra r, seguida de la letra g, seguida de la letra a y termina con la letra r
ciclo	Cadena que comienza con la letra c, seguida de la letra i, seguida de la letra c, seguida de la letra l y termina con la letra o
corto	Cadena que comienza con la letra c, seguida de la letra o, seguida de la letra r, seguida de la letra t y termina con la letra o
decimal	Cadena que comienza con la letra d, seguida de la letra e, seguida de la letra c, seguida de la letra i, seguida de la letra m, seguida de la letra a y termina con la letra l
declaracion	Cadena que comienza con la letra d, seguida de la letra e, seguida de la letra c, seguida de la letra l, seguida de la letra a, seguida de la

Daniel Alejandro Calderón Virgen 18401090
 Luis Sergio Carrillo Sánchez 18401095
 Darien Rafael Márquez Vázquez 17401326
 Diego Arturo Cadena Flores 18401089

	letra r, seguida de la letra a, seguida de la letra c, seguida de la letra i, seguida de la letra o y termina con la letra n
derecha	Cadena que comienza con la letra d, seguida de la letra e, seguida de la letra r, seguida de la letra e, seguida de la letra c, seguida de la letra h y termina con la letra a
detener	Cadena que comienza con la letra d, seguida de la letra e, seguida de la letra t, seguida de la letra e, seguida de la letra n, seguida de la letra e y termina con la letra r
direccionar	Cadena que comienza con la letra d, seguida de la letra i, seguida de la letra r, seguida de la letra e, seguida de la letra c, seguida de la letra c, seguida de la letra i, seguida de la letra o, seguida de la letra n, seguida de la letra a y termina con la letra r.
ejecutar	Cadena que comienza con la letra e, seguida de la letra j, seguida de la letra e, seguida de la letra c, seguida de la letra u, seguida de la letra t, seguida de la letra a y termina con la letra r
encender	Cadena que comienza con la letra e, seguida de la letra n, seguida de la letra c, seguida de la letra e, seguida de la letra n, seguida de la letra d, seguida de la letra e y termina con la letra r
entero	Cadena que comienza con la letra e, seguida de la letra n, seguida de la letra t, seguida de la letra e, seguida de la letra r y termina con la letra o
esperar	Cadena que comienza con la letra e, seguida de la letra s, seguida de la letra p, seguida de la letra e, seguida de la letra r, seguida de la letra a y termina con la letra r
falso	Cadena que comienza con la letra f, seguida de la letra a, seguida de la letra l, seguida de la letra s y termina con la letra o
fijo	Cadena que comienza con la letra f, seguida de la letra i, seguida de la letra j y termina con la letra o
fin	Cadena que comienza con la letra f, seguida de la letra i y termina con la letra n

funcion	Cadena que comienza con la letra f, seguida de la letra u, seguida de la letra n, seguida de la letra c, seguida de la letra i, seguida de la letra o y termina con la letra n
girar	Cadena que comienza con la letra g, seguida de la letra i, seguida de la letra r, seguida de la letra a y termina con la letra r
haz	Cadena que comienza con la letra h, seguida de la letra a y termina con la letra z
iniciar	Cadena que comienza con la letra i, seguida de la letra n, seguida de la letra i, seguida de la letra c, seguida de la letra i, seguida de la letra a y termina con la letra r
izquierda	Cadena que comienza con la letra i, seguida de la letra z, seguida de la letra q, seguida de la letra u, seguida de la letra i, seguida de la letra e, seguida de la letra r, seguida de la letra d y termina con la letra a
largo	Cadena que comienza con la letra l, seguida de la letra a, seguida de la letra r, seguida de la letra g y termina con la letra o
mientras	Cadena que comienza con la letra m, seguida de la letra i, seguida de la letra e, seguida de la letra n, seguida de la letra t, seguida de la letra r, seguida de la letra a y termina con la letra s
nulo	Cadena que comienza con la letra n, seguida de la letra u, seguida de la letra l y termina con la letra o
no	Cadena que comienza con la letra n y termina con la letra o
pin	Cadena que comienza con la letra p, seguida de la letra i y termina con la letra n
principal	Cadena que comienza con la letra p, seguida de la letra r, seguida de la letra i, seguida de la letra n, seguida de la letra c, seguida de la letra i, seguida de la letra p, seguida de la letra a y termina con la letra l
retorno	Cadena que comienza con la letra r, seguida de la letra e, seguida de la letra t, seguida de la letra o, seguida de la letra r, seguida de la letra n y termina con la letra o
si	Cadena que comienza con la letra s y termina con la letra i

verdadero	Cadena que comienza con la letra v, seguida de la letra e, seguida de la letra r, seguida de la letra d, seguida de la letra a, seguida de la letra d, seguida de la letra e, seguida de la letra r y termina con la letra o
direccionar	Cadena que comienza con la letra d, seguida de la letra i, seguida de la letra r, seguida de la letra e, seguida de la letra c, seguida de la letra c, seguida de la letra i, seguida de la letra o, seguida de la letra n, seguida de la letra a y termina con la letra r
girar	Cadena que comienza con la letra g, seguida de la letra i, seguida de la letra r, seguida de la letra a y termina con la letra r
seguir	Cadena que comienza con la letra s, seguida de la letra e, seguida de la letra g, seguida de la letra u, seguida de la letra i y termina con la letra r
arriba	Cadena que comienza con la letra a, seguida de la letra r, seguida de la letra r, seguida de la letra i, seguida de la letra b y termina con la letra a
leer	Cadena que comienza con la letra l, seguida de la letra l, seguida de la letra l y termina con la letra r
imprimir	Cadena que comienza con la letra i, seguida de la letra m, seguida de la letra p, seguida de la letra r, seguida de la letra i, seguida de la letra m, seguida de la letra i y termina con la letra r
video	Cadena que comienza con la letra v, seguida de la letra i, seguida de la letra d, seguida de la letra e y termina con la letra o
parav	Cadena que comienza con la letra p, seguida de la letra a, seguida de la letra r, seguida de la letra a y termina con la letra v
ruta	Cadena que comienza con la letra r, seguida de la letra u, seguida de la letra t y termina con la letra a

Tipos de datos

Tipo	Descripción	Tamaño	Rango	Ejemplo
booleano	Verdadero o falso	1 bit	0 ó 1	Verdadero
corto	Entero pequeño	8 bits	-128 a 127	110
carácter	Elemento de la tabla ASCII	8 bits	0 a 255	'c', 'l', '3', 65
entero	Entero	16 bits	-32568 a 32567	-3332
largo	Entero largo	32 bits	-2,147,483,648 a 2,147,483,647	-100000, 2000000
decimal	Punto flotante	32 bits	10e-38 a 10e38	3.1416, 99.9999
cadena	Texto plano	64 bytes	0-64 caracteres	'Hola'

Cadenas

En **LYADROME** la forma en que se ponen las cadenas es usando las comillas simples '. Si la cadena tiene un solo carácter se seguirá tratando como un tipo de dato cadena y no como un tipo de dato carácter y no marcará error, sin embargo, una variable de tipo carácter que se definen también se definen con comillas simples, que tenga más de dos caracteres marcará error, ya que sobrepasará el espacio que está reservado para esa memoria.

Una cadena **no** puede contener caracteres que no estén definidos en el alfabeto tales como vocales con tilde, la letra ñ, o caracteres que se usan en otros lenguajes diferentes al español.

Números

Los números se pueden expresar en tres formas diferentes: como número entero, decimal o con notación científica

Ejemplos:

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

- **Enteros** 3,5,100,129,2049
- **Decimal** 3.1416, 99.99, 30.0000001
- **Notación científica:** 10e-7, 43.3e9, 8e30

Operadores

tipo	nombre	símbolo(s)
Aritméticos	suma	+
	resta	-
	división	/
	multiplicación	*
	módulo	%
Relacionales	igual	==
	mayor que	>
	menor que	<
	mayor o igual que	>=
	menor o igual que	<=
	desigual	<>
Lógicos	and	&
	or	
	not	!

	xor	\wedge
Otros	incremento	++
	decremento	--
	potencia	**

Identificadores

Un identificador es el nombre que le daremos a variables y funciones que utilizaremos. Para poder definir este nombre tenemos que tener en cuenta que debe de comenzar con una letra minúscula de la [a - z] seguido de 0 a 31 caracteres más, los cuales pueden ser letras de la [a - z] y dígitos de [0 - 9] además de _un guion bajo [_].

Ejemplos de identificadores incorrectos

3estado	numero-1	dirección	3435
---------	----------	-----------	------

Ejemplo de identificadores correctos

estado	numero1	direccion	trescuatrotrescinco	doge_coin_moon
--------	---------	-----------	---------------------	----------------

Variables

La variable es la unidad básica de almacenamiento en un programa y las cuales servirán de apoyo durante la ejecución del programa. Una variable se define mediante la combinación de un identificador, un tipo y un inicializador. Para poder inicializar una variable se utiliza el carácter de igual '=' y seguido se pone el valor correspondiente al tipo de dato. Y como cosa importante, las variables solo se pueden declarar en la parte de *declaración* la cual estará en la parte inicial del código, esto también incluye a las variables que se van a usar en las funciones.

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

Es importante mencionar que cada variable tiene que estar declarada en una línea, no se pueden declarar múltiples variables en una sola línea especificando sólo el tipo de dato.

Formato tipo_de_dato nombre_de_la_variable = valor_de_la_variable

Ejemplo: **entero** tiempo = 0

Arreglos

Un arreglo es un grupo de variables del mismo tipo al que se hace referencia por medio de un nombre que los identifica, para declarar un arreglo en **LYADRON** se tiene que poner la palabra reservada **arreglo** después de poner el tipo de dato. Para indicar el tamaño del arreglo se puede hacer de dos formas: inicializando los elementos separados por una coma o indicando el tamaño entre paréntesis con un número. Si se hace de la segunda manera los elementos se inicializan automáticamente según como corresponda con su tipo de dato.

Formato: tipo_de_dato nombre **arreglo** = (tamaño)

tipo_de_dato nombre **arreglo** = elemento1, elemento2,...elementoN

Ejemplo: **entero** coordenadas **arreglo** = 33,54,67,32

entero coordenadas **arreglo** = (4)

Funciones

Las funciones son una manera de ahorrar código, ya que se puede escribir en ella alguna cosa que se puede repetir en diferentes partes del código. Para definir una función se tiene que indicar por medio de la palabra **funcion** seguido del nombre que al ser un identificador tiene que seguir la regla de ellos. Una función puede recibir parámetros, los cuales se especifican entre paréntesis después del nombre anteponiendo el tipo de datos, Si la función va a retornar un valor, se tiene que indicar de qué tipo, si no, no se pone nada. Para indicar que la función va a iniciar se ponen dos puntos y para terminar se indica con la palabra fin. Si la función va a retornar un valor, se usa la palabra reservada **retornar** seguido de la variable o valor

Formato **funcion** nombre_de_la_funcion (tipo_de_parametro1 parametro1,

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

```

                                tipo_de_parametro2 parametro2,..
                                tipo_de_parametroN
                                parametroN,)tipo_de_valor_de_retorno:
    cuerpo de la función
    retorno variable o valor_de_retorno;

fin

```

Ejemplo **funcion** areaTriangulo(**entero** base, **entero** altura)**entero**:

```

    retorno base*altura/2;

fin

```

Llamar una función

Para mandar a llamar una función se usa la palabra reservada ***ejecutar*** seguido del nombre de la función con sus parámetros entre paréntesis y separados por una coma en de que la función los requiera. Si una función retorna algún valor, este puede usarse para asignarse a una variable si y sólo si los tipos de datos son iguales.

Ejemplo: ruta = ***ejecutar*** buscarSiguieteParada();

Fin de una sentencia

Para diferenciar entre una sentencia y otra en LYADRON se usa el comúnmente conocido punto y coma ‘;’, de esta forma para el programador será más sencillo identificar una sentencia y otra, y claro, también por otra parte sirve de gran ayuda para el compilador a la otra de estar trabajando.

Comentarios

Para que el lenguaje pueda distinguir qué secciones del código fuente son un comentario esta líneas o bloques deben de incluir el doble - para una línea de código y triple - con inicio y fin para un bloque completo de código (donde solo se considera lo que está entre el inicio y el final). Por ejemplo:

```

--Este es un comentario en una solo línea de código

---

```

Este es un bloque de comentarios

Condicionales

si

El **si** es un controlador de flujo que se usa mucho en cualquier lenguaje de programación donde es comúnmente conocido como *if*, con el *si* se evalúa una condición la cual en caso de que sea verdadera ejecutará el bloque de código que estará dentro. La condición que se evalúa puede expresarse mediante variables booleanas o usando operadores relacionales que comparan números o caracteres. Habrá ocasiones en que se tenga que evaluar dos condiciones, para ello se pueden usar los operadores lógicos con los cuales se pueden ahorrar un *si*.

La forma en que se debe escribir un *si*, es iniciando con la palabra reservada **si** seguido de entre paréntesis la condición o condiciones, después los dos puntos para indicar que iniciara el bloque de código y para terminar se pone la palabra fin.

Formato **si**(condición):

bloque de código

fin

Formato **si**(carga<10):

mensajeAdvertencia();

fin

si no

Cuando se usa el *si* dependiendo del resultado de la expresión se va ejecutar el código dentro o saltara a la línea que hay después del fin. Pero si dependiendo de la condición se requiere hacer algo en específico no se tiene que hacer otro *si*, ya que se puede especificar por medio de las palabras reservadas **si no** que en cambio se ejecute otro bloque de código, de esta manera se estará tratando la dicotomía haciendo una cosa u otra. La forma en que se usa es poniendo **si no** antes del *fin* del *si*, siguiendo de los dos puntos para indicar que el bloque comenzará.

Formato **si**(condición):

bloque de código

si no:

bloque de código

fin

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

Ejemplo **si**(ruta == fin):
 empezarAterrizaje()
 si no:
 buscarSiguienteParada()
 fin

Si y si no anidados

La anidación es muy común en los lenguajes de programación, para hacerlo en LYADRON solo debe de tenerse en cuenta la ubicación del fin que le corresponde al *si* y al *si no*. Lo recomendable es tener todo bien indentado por bloques para que no se mezclen las sentencias de cada bloque

Ejemplo **si**(condición):
 bloque de código
 si no:
 si(condición):
 bloque de código
 si no:
 si(condición):
 bloque de código
 si no:
 bloque de código
 fin
 fin
 fin
 fin

Ciclos

ciclo

La palabra reservada *ciclo*, hace referencia a lo que comúnmente se conoce como el ciclo *for*, este ciclo permite repetir varias instrucciones un determinado número de ocasiones según las condiciones que se le den. La forma en la que utilizaremos este ciclo será la siguiente:

```
ciclo (inicialización; condición; incremento/decremento)
    instrucciones;
fin
```

En la sección de condición se pueden utilizar variables de tipo entero previamente definidas en el programa.

Ejemplo:

```
entero suma=0
ciclo (i=1;i<=100;i++)
    suma=suma+i
fin
```

mientras

La palabra reservada *mientras* hace referencia a lo que comúnmente se conoce como el ciclo *while*, este permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor verdadero). La forma en la que utilizaremos este ciclo será la siguiente:

```
mientras (condición)
    sentencia;
fin
```

De igual forma en la condición se pueden utilizar variables de tipo entero o tipo booleana previamente definidas en el programa.

Ejemplo:

```
entero i=1
entero suma=0
mientras(i<=100)
    suma=suma+i;
    i++;
fin
```

haz - mientras

Un ciclo **haz-mientras** tiene la ventaja de que, si la condición que se evalúa en el **mientras** es falsa, al menos una vez el bloque de código es ejecutado.

La forma de usarlo sería la siguiente.

```
haz:
    bloque de código
mientras (condición)
fin
```

De igual forma en la condición se pueden utilizar variables de tipo entero o tipo booleana previamente definidas en el programa:

Ejemplo

```
entero suma = 0

haz:

    suma=suma+5

mientras (suma < 50)

fin
```

Constantes del lenguaje

Una constante en **LYADRON** es una palabra reservada que le brinda al programador facilidades al momento de estar haciendo código. Estas constantes son llamadas anteponiendo en ellas el @ seguido de los elementos que dichas constantes requieren.

apagar

Con la palabra *apagar* se empiezan a detener las hélices del dron.

encender

Con la palabra *encender* se empiezan a funcionar las hélices.

calibrar

Con esta palabra se calibran las hélices, es recomendable siempre usar esta palabra al iniciar para actuar dicha función.

direccionar

Haciendo uso de la palabra *direccionar*, el dron se moverá en la dirección que se especificada. Esta palabra al ser un verbo, siempre va acompañada de otra palabra que indicara la dirección.

@*direccionar* @direccion;

arriba

arriba es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

abajo

abajo es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

izquierda

izquierda es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

derecha

derecha es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

adelante

adelante es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

atras

atras es una constante del lenguaje la cual indica la dirección en que se va a mover el dron.

Como el efecto que tendrán estas palabras se verán reflejadas directamente la dirección que tome el dron su uso solamente en conjunto con la palabra **@*direccionar*** y las posibles combinaciones de movimiento serán dadas por alguna función. Ejemplo:

***mientras* (variable1!=0)**

@*direccionar* @derecha

fin

Como se menciona anteriormente estas palabras solo se pueden usar dentro de alguna función esto se hace así porque si se usa fuera de ellas el lenguaje no sabría reconocer que se quiere tomar esa dirección en específico.

girar

Con la palabra girar se podrá *girar* hacia un lado con una inclinación especificada con un número que estará expresado en grados, esta es una alternativa a la palabra *direccionar*.

@girar 45

fijo

La palabra reservada fijo indica al drone que debe mantenerse volando sin hacer ningún cambio de dirección esto significa que simplemente se mantendrá suspendido en la posición en la que está. La forma en la que se utiliza es la misma que en las palabras reservadas de dirección vistas en el subtema anterior.

acelerar

Para poder aumentar o bajar la velocidad se puede hacer uso de la palabra reservada acelerar, la cual incrementara la velocidad en que avanza el dron si se pone un 1 o decrementar si se pone un 0.

detener

La palabra indica al drone que debe dejar de moverse y tiene que comenzar a descender hasta que llega al suelo, esta palabra reservada es la ideal para poder finalizar cualquier cosa que el drone está haciendo y en el lugar donde se encuentre simplemente baje. La forma en la que se utiliza es la misma que en las palabras reservadas de dirección vistas en el subtema anterior.

esperar

Como parte de las palabras reservadas del lenguaje está la palabra *esperar*, con la cual por medio de ella la siguiente línea será ejecutada después del tiempo que se especifique en milisegundos, el cual se indica después de poner la palabra

Ejemplo: **@esperar** 1000;

seguir

Con la palabra reservada *seguir* el dron funcionara como en piloto automático, para indicar el tiempo que durara así se tiene que especificar por medio de un número, el cual también estará expresado en milisegundos.

Ejemplo: **@seguir** 1000

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

pin

Dentro de las palabras reservadas en **LYADROME** encuentra la palabra *pin*, la cual hará referencia a un pin en específico del microcontrolador. La forma de usarla es poniendo el carácter **@pin** seguido de un número que indicará la patilla seguido de la palabra reservada **@apagar** o **@encender**

Ejemplo **@pin 18 @encender**

leer

La instrucción leer permite la lectura de valores y caracteres en una terminal de texto o dispositivo de entrada, su forma de uso es, como otras instrucciones utilizando el carácter '@' seguido de la palabra reservada **leer** seguida a su vez de un identificador de variable previamente definida.

Ejemplo **@leer a**

imprimir

Con la palabra reservada imprimir se permite la visualización de valores dentro de un identificador en una terminal o dispositivo de salida, para utilizarla se utiliza el carácter '@' seguido de la palabra reservada imprimir seguida a su vez del identificador a visualizar.

Ejemplo **@imprimir a**

captura

Con esta palabra reservada indicamos, en caso de tener disponible una cámara tomar una fotografía. Para utilizar esta función se utiliza el carácter '@' seguido de la palabra reservada captura.

Ejemplo **@captura**

video

Con esta palabra reservada indicamos, en caso de tener disponible una cámara tomar una fotografía. Para utilizar esta función se utiliza el carácter '@' seguido de la palabra reservada captura.

Ejemplo **@video**

parav

Con esta palabra reservada se detiene la grabación, para que esta palabra pueda usarse debe de estar antes la palabra reservada **@video** la cual indicara que hay una grabación, y en caso de que no se use antes marcara un error, ya que no hay grabación que detener.

Ejemplo **@parav**

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089

ruta

Esta palabra reservada se utiliza para definir una ruta marcada por pares ordenados. Esta ruta la seguirá de forma consecutiva desplazándose en línea recta y por orden de un punto a otro. La forma en la que se toman los puntos es como en un plano cartesiano donde el lugar de partida del dron será el mismo de la posición (0,0).

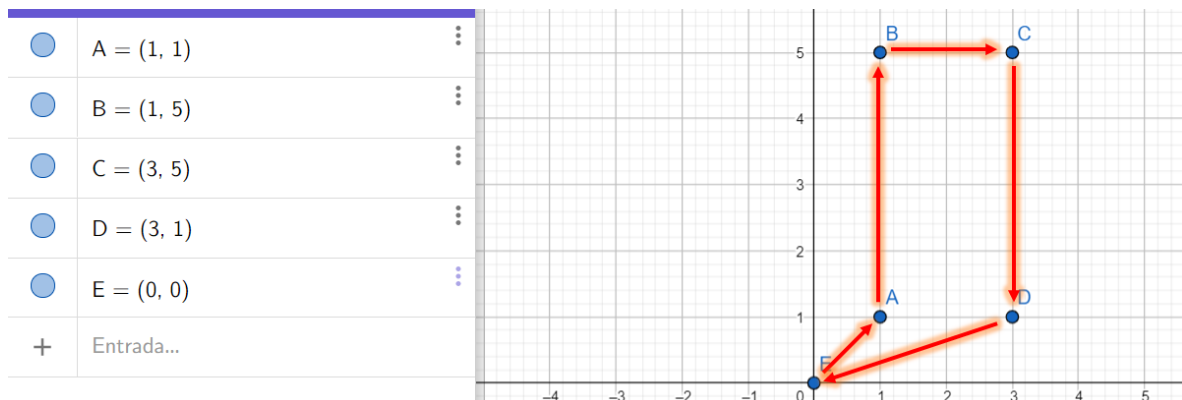
Los puntos se indicarán poniéndolos dentro de un paréntesis inmediatamente después de la palabra reservada, el formato dentro del paréntesis será como puntos ordenados separados por comas, en donde cada par de números se tomará como (x,y).

La sintaxis completa para sería la siguiente:

@ruta(x,y,x,y,x,y)

Ejemplo

@ruta(1,1,1,5,3,5,3,1,0,0)



Aquí el punto de partida será el 0.

La unidad de medida por defecto será en metros y esta no se podrá modificar, por lo que se tendrá que realizar las conversiones necesarias e indicarlo de esta forma. Se pueden poner números negativos y decimales.

Formato del script

La forma en que se puede escribir código usando **LYADROME** tiene que ser de una forma en que todo debe de tener su inicio y su fin, los cuales se expresan con los dos puntos ':' y la palabra fin

respectivamente. Y estará dividido en cuatro partes: una parte opcional para cargar librerías, la declaración de variables, la parte principal y la parte de funciones la cual también es opcional.

Para englobar estas cuatro partes se comienza con la palabra ***iniciar*** seguido del nombre del archivo y los dos puntos y en la parte final la palabra ***fin***.

Ejemplo ***iniciar*** Archivo:
código

fin

Después de indicar el inicio del script se ponen las librerías que se van a usar, y en dado caso que no sean necesarias esta parte se puede omitir

La parte de la declaración es la segunda que se pone, en ella se pondrán todas las variables que se utilizarán.

Ejemplo ***declaracion:***
variable 1
variable 2
.....
variable N

fin

En la parte principal se incluyen toda la lógica que se va a usar,

Ejemplo ***principal:***
código

fin

Si se incluirán funciones estas se deberán poner en la parte después de la principal y antes del último fin, se pueden escribir todas las funciones que sean necesarias.

Ejemplo ***funcion:***
código

fin

Un script de ***LYADRON***E juntando las partes anteriores tendrá la siguiente estructura.

iniciar Archivo:
@cargar librería

Daniel Alejandro Calderón Virgen 18401090
Luis Sergio Carrillo Sánchez 18401095
Darien Rafael Márquez Vázquez 17401326
Diego Arturo Cadena Flores 18401089

```

    declaracion:
        variable 1
        variable 2
        .....
        variable N
    fin
principal:
    código
fin

    funcion1:
        código
    fin
    .....
    funcionN:
        código
    fin
fin

---
```

Cargar librerías

En **LYADROME** la que se pueden agregar librerías es usando la palabra reservada **cargar** seguido del nombre de dicha librería, estas librerías contendrán funciones que harán que la programación sea más sencilla.

El lugar en donde se mandan a llamar las librerías es después de iniciar el código, antes de las declaraciones de las variables

Ejemplo de código

iniciar Ejemplo:

```

    declaracion:
        entero contador = 0;
    fin

    principal:
        ejecutar parpadear();
    fin

    funcion parpadear():
        mientras(contador<1000)
            @esperar 1000;
            si(contador%2 == 0):
```

Daniel Alejandro Calderón Virgen 18401090

Luis Sergio Carrillo Sánchez 18401095

Darien Rafael Márquez Vázquez 17401326

Diego Arturo Cadena Flores 18401089


```

                                ejecutar encender();
                        si no:
                                ejecutar apagar();
                        fin
                fin
        fin

funcion encender():
        @pin 13 @encender;
        fin

funcion apagar():
        @pin 13 @apagar;
        fin
fin

```

Bibliografía

Schildt, H. (2000). Java Manual de referencia. 7ma. McGRAW-HILL. Obtenido de <https://usermanual.wiki/Document/Java20Manual20de20Referencia2C207ma20Edici3B3n2020Herbert20SchildtFREELIBROSORG.157528914>

V, R. cout and cin in C++ - CodeSpeedy. Retrieved 17 May 2021, from <https://www.codespeedy.com/cout-and-cin-in-cpp/>