

Introducción a la programación de microcontroladores con Tinkercad

Cohetes de agua como recurso educativo:
desde la motivación científico-tecnológica en aula hasta la participación del
alumnado en un concurso en Castellón

**WATER
ROCKETS**



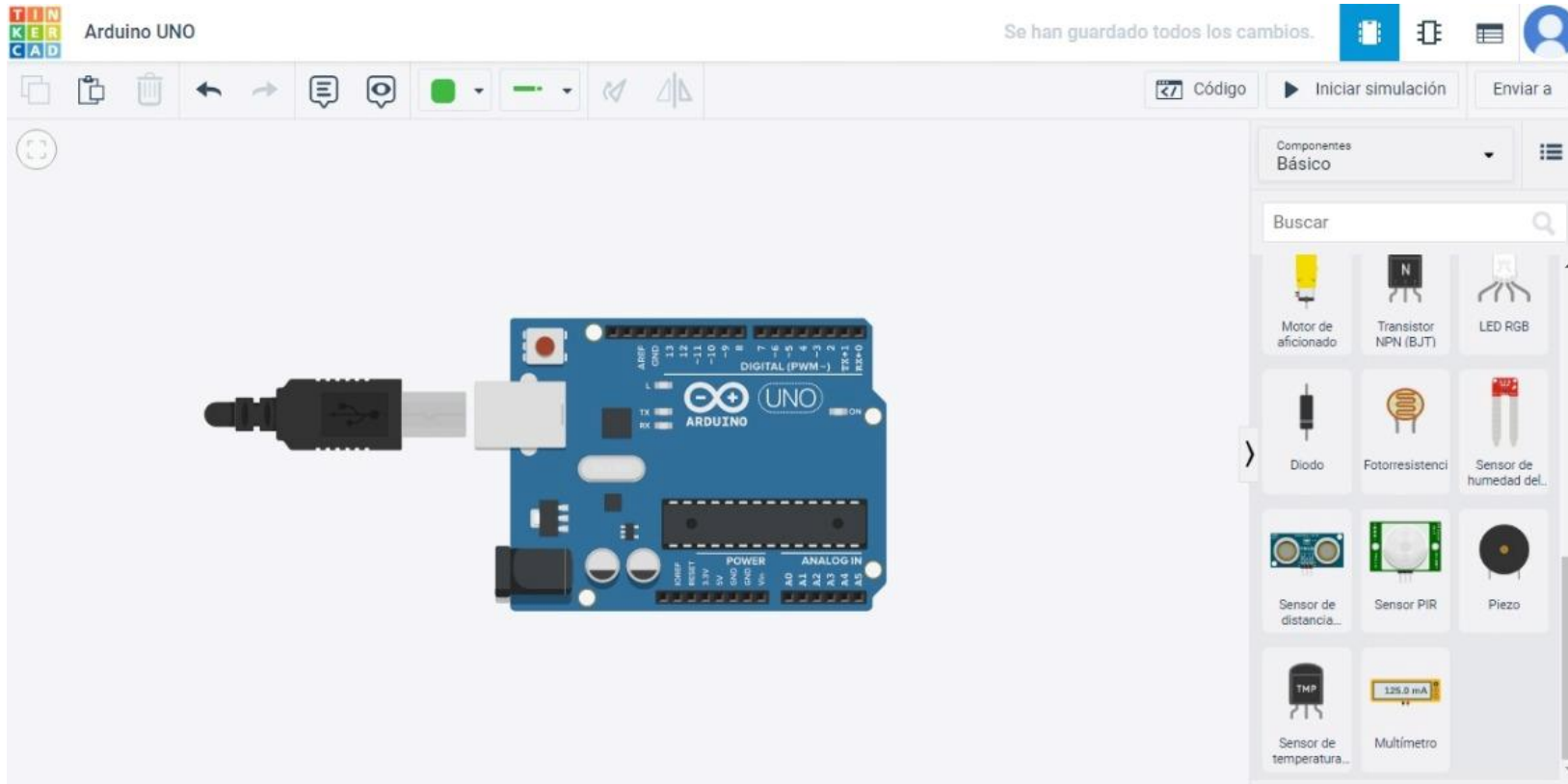
Cohetes de agua como recurso educativo: desde la motivación científicotecnológica hasta la participación en un concurso

¿Qué es Tinkercad?

- Tinkercad es un software gratuito para diseño y modelado 3D.
- Es un software intuitivo que ofrece una gran biblioteca de formas prediseñadas. Su interfaz pensada para estudiantes.
- Se pueden exportar los diseños a STL e imprimir con impresora 3D.
- Incluye un simulador de circuitos con placas Arduino.
- Programación por bloques para niños.

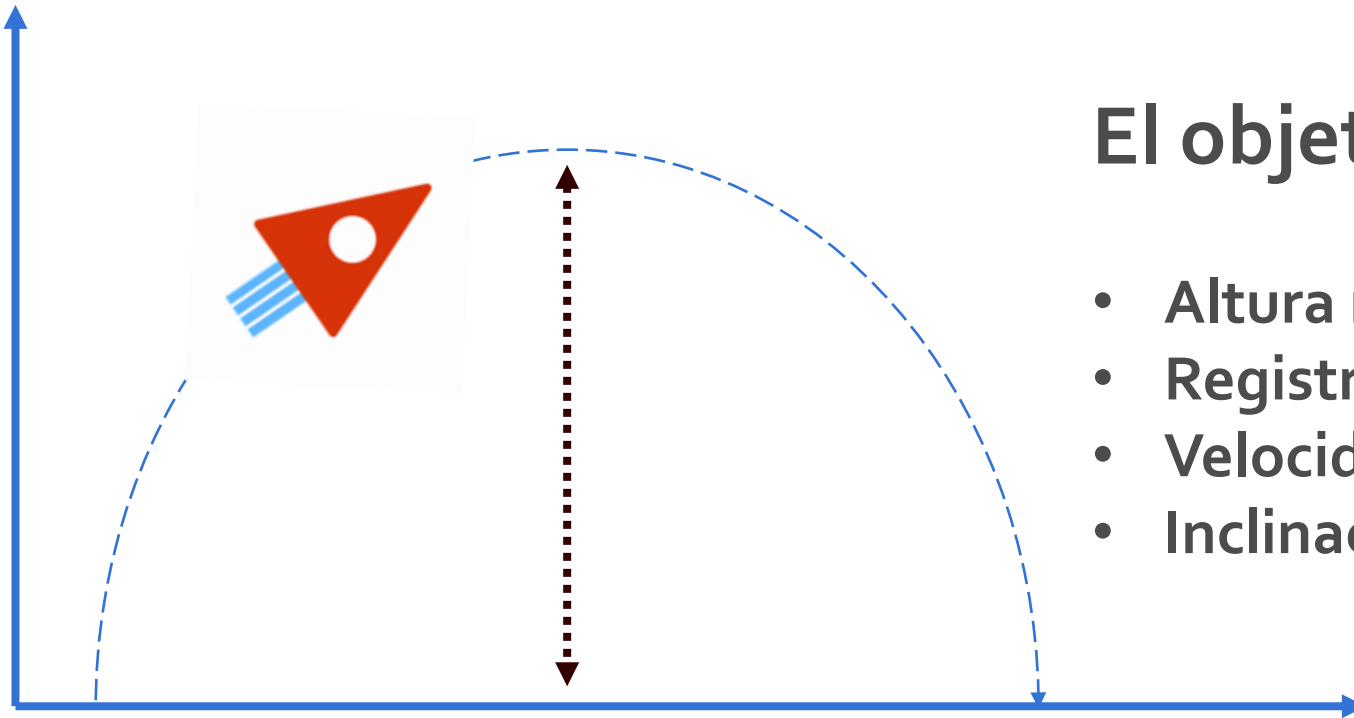


Simulador de circuitos



www.tinkercad.com

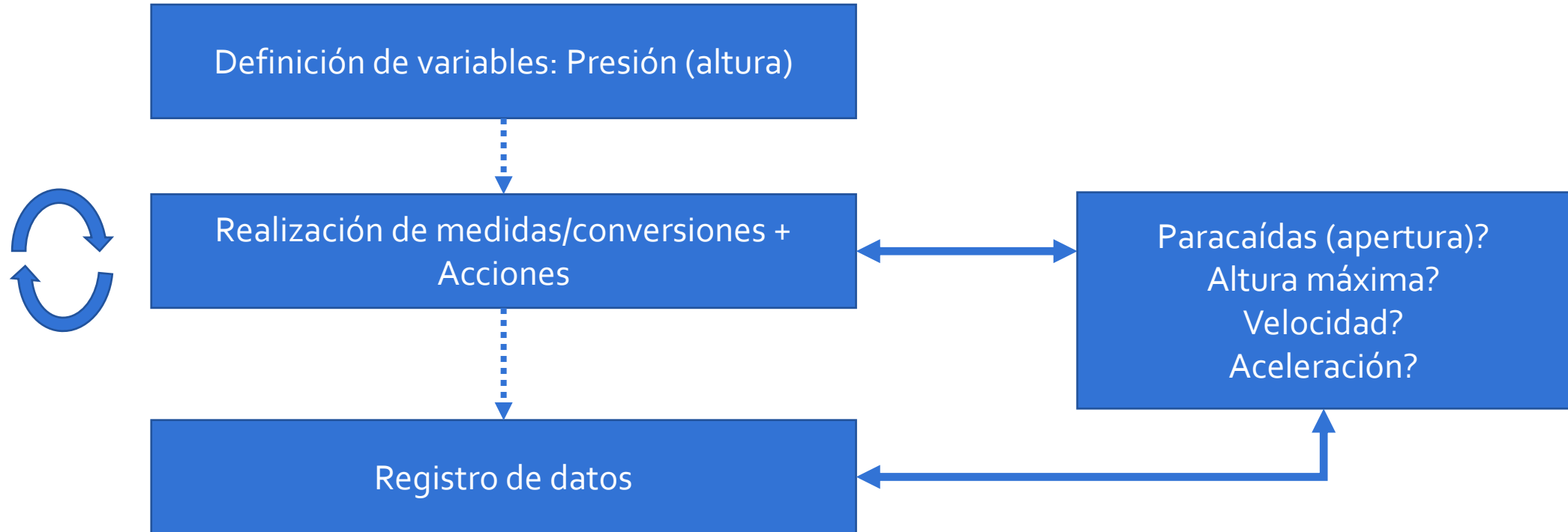
Programa ejemplo telemetría



El objetivo?:

- Altura máxima?
- Registro trayectoria?
- Velocidades/aceleraciones?
- Inclinación?

Programa ejemplo telemetría



Programa ejemplo telemetría

Variables

```
int i, h[40];  
char valor;
```

Configuraciones iniciales

```
void setup()  
{  
  Serial.begin(9600);  
}
```

Lectura datos

```
void loop()  
{  
  if(Serial.available()){  
    valor=Serial.read();  
  
    if(valor=='1'){  
      Serial.println("Tomando datos...");  
      for(i=0;i<40;i=i+1){  
        h[i]=analogRead(A0);  
        delay(200);  
      }  
      Serial.println("Todos los datos tomados");  
    }  
  }  
}
```

Mostrar datos

```
if(valor=='2'){  
  for(i=0;i<40;i=i+1){  
    Serial.println(h[i]);  
  }  
}
```

Operaciones

```
if(valor=='3'){  
  int max=0;  
  for(i=0;i<40;i=i+1){  
    if(h[i]>max) max=h[i];  
  }  
  Serial.println("Altura max");  
  Serial.println(max);  
}  
}
```

Programación en Arduino

Lenguaje similar “C++”

- Tipos de datos
- Estructuras de programación: bucles, condicionales,...
- Comunicación entre dispositivos
- Entradas y salidas disponibles: digital y analógico

Ejemplos:

1. Lectura/escritura de señales digitales: LED
2. Comunicación por puerto Serial
3. Lectura de /escritura de señales analógicas: Potenciómetro
4. ...



Cohetes de agua como recurso educativo: desde la motivación científicotecnológica hasta la participación en un concurso

Tipos de datos

- Para crear una variable en C se debe especificar el tipo de variable y después el nombre con el que lo identificaremos.
- Por ejemplo para crear una variable con un valor entero (int):

int variable;

Tipos de Datos	Memoria que ocupa	Rango de valores
boolean	1 byte	0 o 1 (True o False)
byte/ unsigned char	1 byte	0 – 255
char	1 byte	-128 – 127
int	2 bytes	-32768 – 32767
word / unsigned int	2 bytes	0 – 65535
long	4 bytes	-2147483648 – 2147483647
unsigned long	4 bytes	0 - 4294967295
float / double	4 bytes	-3,4028235E+38 - 3,4028235E+38
string	1 byte + x	Array de caracteres
array	1 byte + x	Colección de variables

Instrucciones condicionales

- La instrucción *if* condicional la ejecución en función de si una operación da como resultado *true* o *false*.
- En caso de que no se cumpla, se ejecuta la instrucción *else*.
- Se pueden añadir más condicionales con la instrucción *elseif*, en el caso que se cumpla una, no se comprueban el resto.
- Ejemplos de instrucciones

`x == 2`

`x != 2`

`x < 5`

`x > 5`

`x >= 5`

`x <= 5`

`x < 3 || x > 5`

`x <= 3 && r == 7`

```
if(someCondition) {  
    // do something  
} else if {  
    // do something  
} else if {  
    // do something  
} else {  
    // do something  
}
```

Bucles (for)

- Para repetir la ejecución de unas líneas de código se utilizan bucles. En el bucle *for* se introduce:
 - Se crea el valor la variable y se inicializa su valor inicial (*int i = 0*)
 - La condición para repetir el bucle (*i < 5*)
 - La actualización de la variable en cada iteración (*i = i+1*)
- Por ejemplo:

```
for(i = 0; i < 5; i = i + 1) {  
...  
}
```

- En el caso anterior, se repetiría el código entre corchetes 5 veces, al finalizar el bucle, el valor de *i* sería 5.

Partes de un programa de Arduino

```
void setup() {  
  // put your setup code here, to run once:  
  
}
```

Inicialización

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Ejecución continua del código

Comunicación serie

- Inicializamos el puerto serie (9600 son los bits por segundo, la velocidad de comunicación)

```
Serial.begin(9600);
```

- Para imprimir un valor por el puerto serie.

```
Serial.println("Hola");
```

Escritura!

- **Print (continuo)**
- **Println (retorno de carro)**

- Para leer un valor del puerto serie, se debe primero comprobar que haya información disponible en el puerto serie. Devuelve 1 si hay información.

```
Serial.available();
```

- Para leer el valor del puerto serie.

```
Serial.read();
```

Lectura!

Ejemplo 1: Arduino contesta

Realizar un código, que haga que el Arduino envíe lo que le enviamos por puerto serie.

Pasos:

1. Inicializar puerto serie sólo una vez (*Serial.begin(9600)*).
2. Después en cada iteración comprobar si hay datos en el puerto serie (*Serial.available()*).
3. Si hay datos disponibles, leer los datos (*Serial.read()*) y si estos son igual a '1', enviar por puerto serie "Hola".

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    if(Serial.available()){
        if(Serial.read()=='1'){
            Serial.print("Hola");
        }
    }
}
```

Ejemplo 2: Operar números

Crear dos variables e inicializar su valor, mostrar por puerto serie el resultado de sumarlos, restarlos, multiplicarlos y dividirlos.

Pasos:

1. Crear un *int* *x* y un *double* *y*. Inicializar las variables con los valores que se quiera.
2. Mostrar por puerto serie los resultados. (*Serial.println(x+y)*).
3. Esperar 5 segundo antes de volver a enviar por puerto serie los resultados.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int x;
  double y;

  x = 4;
  y = 3.2;

  Serial.println(x+y);
  Serial.println(x-y);
  Serial.println(x*y);
  Serial.println(x/y);
  Serial.println( );

  delay(5000);
}
```

Entradas y salidas digitales

- Los pines digitales pueden ser salidas o entradas. Para configurarlos en el *setup()* se debe utilizar la instrucción (en la cual se configura el pin digital 2 como salida):

pinMode(2, OUTPUT) o pinMode(2, INPUT)

- Para leer un pin digital se utiliza la instrucción:

digitalRead(2)

- Para escribir en un pin digital se utiliza:

digitalWrite(2, LOW) o digitalWrite(2, HIGH)

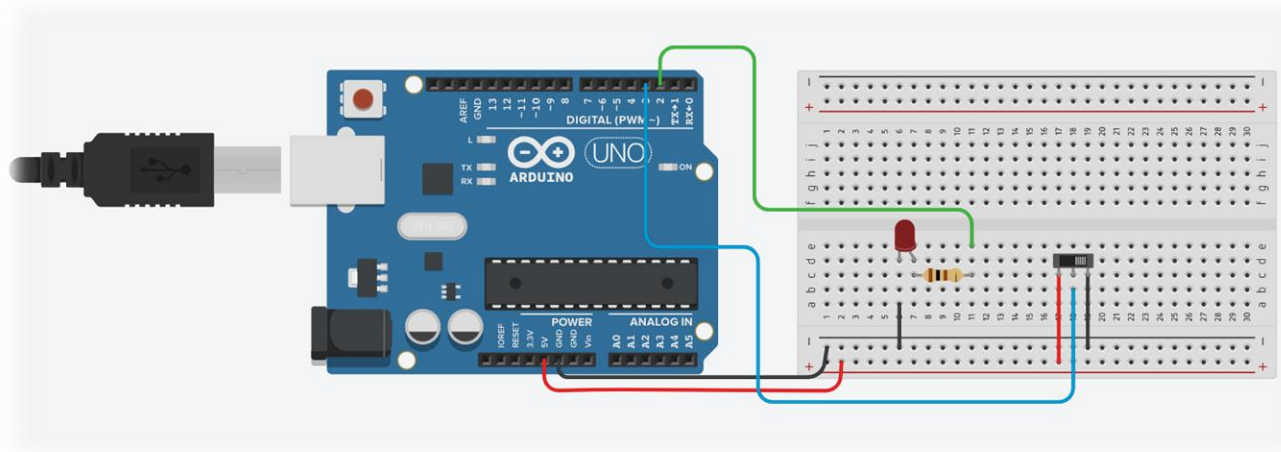
digitalWrite(2, 0) o digitalWrite(2, 1)

Ejemplo 3: Parpadear LED

Crear un código, que haga parpadear el LED de forma controlada por el usuario.

Pasos:

1. Establecer el modo de los pines (pinMode)
2. Crear una condición basada en una entrada digital. Se controlará el valor de esta entrada mediante un interruptor.
3. Utilizar la función *delay(1000)* para temporizar la activación del LED (parpadeo).



Ejemplo 3: Parpadear LED

Crear un código, que haga parpadear el LED de forma controlada por el usuario.

Pasos:

1. Establecer el modo de los pines (pinMode)
2. Crear una condición basada en una entrada digital. Se controlará el valor de esta entrada mediante un interruptor.
3. Utilizar la función *delay(1000)* para temporizar la activación del LED (parpadeo).

```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
}

void loop()
{
  if(digitalRead(3)) {
    digitalWrite(2, 1);
    delay(500);
    digitalWrite(2, 0);
    delay(500);
  }
  else
    digitalWrite(2, 0);
}
```

Entradas analógicas

- Las entradas analógicas son A1, A2, A3, A4 y A5 en el Arduino UNO. Para leer el valor por ejemplo de A2:

analogRead(A2)

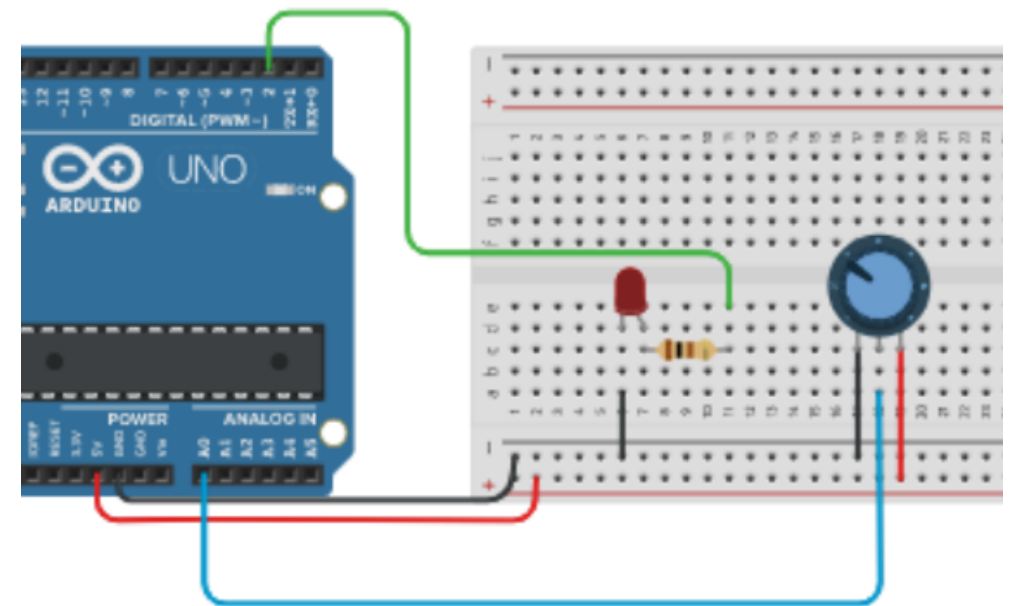
- Devuelve un valor entero de 0 a 1023 (10 bits) dependiendo de la referencia de voltaje del Arduino.
- Para obtener el voltaje, conociendo el voltaje de referencia en el Arduino, se debe hacer una regla de tres, ya que la relación es lineal.

Ejemplo 4: Leer entrada analógica

Hacer un código, que lea una entrada analógica y devuelva el valor en voltios por puerto serie.

Pasos:

1. Hacer un circuito con un potenciómetro y conectarlo a una entrada analógica.
2. Enviar por puerto serie el valor del voltaje con decimales.
3. Añadir un retraso de 0.5 segundos, es decir, que se envíe el valor cada medio segundo.



Ejemplo 4: Leer entrada analógica

Hacer un código, que lea una entrada analógica y devuelva el valor en voltios por puerto serie.

Pasos:

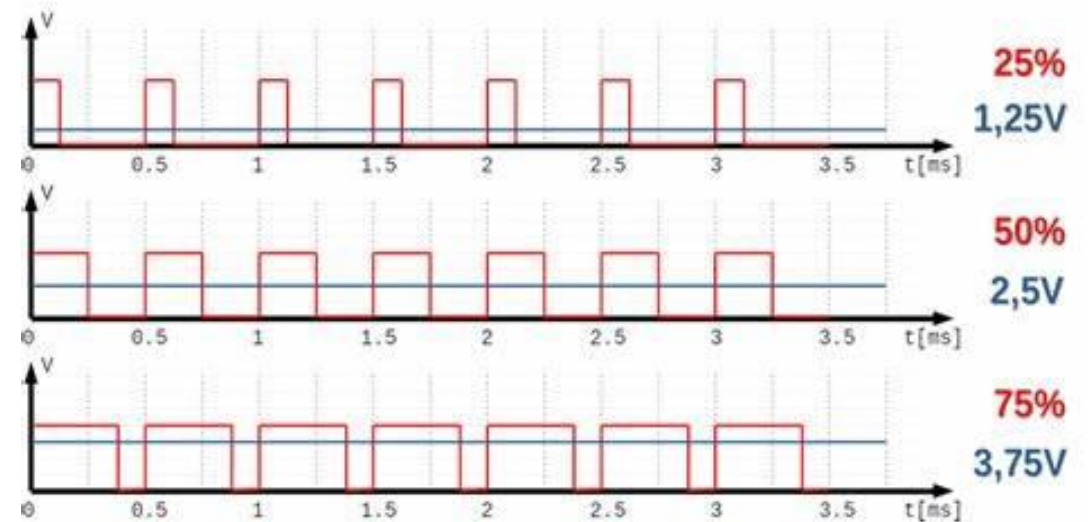
1. Hacer un circuito con un potenciómetro y conectarlo a una entrada analógica.
2. Enviar por puerto serie el valor del voltaje con decimales.
3. Añadir un retraso de 0.5 segundos, es decir, que se envíe el valor cada medio segundo.

```
int valor;  
float voltaje;  
  
void setup()  
{  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  valor = analogRead(A0);  
  voltaje = valor*5/1023.0;  
  Serial.println(voltaje);  
  delay(500);  
}
```

Salidas analógicas

- Las salidas analógicas funcionan por la técnica pulse-width modulation (PWM). Se modula el ancho del pulso de señales digitales para modificar el voltaje medio.
- Con un pequeño condensador, se tendría una señal continua, eliminando la frecuencia del pulso.
- Sólo algunos pines digitales del Arduino soportan esta técnica.
- Ejemplo: para poner el pin 2 a un valor de tensión de 5 voltios.

analogWrite(2,1023)

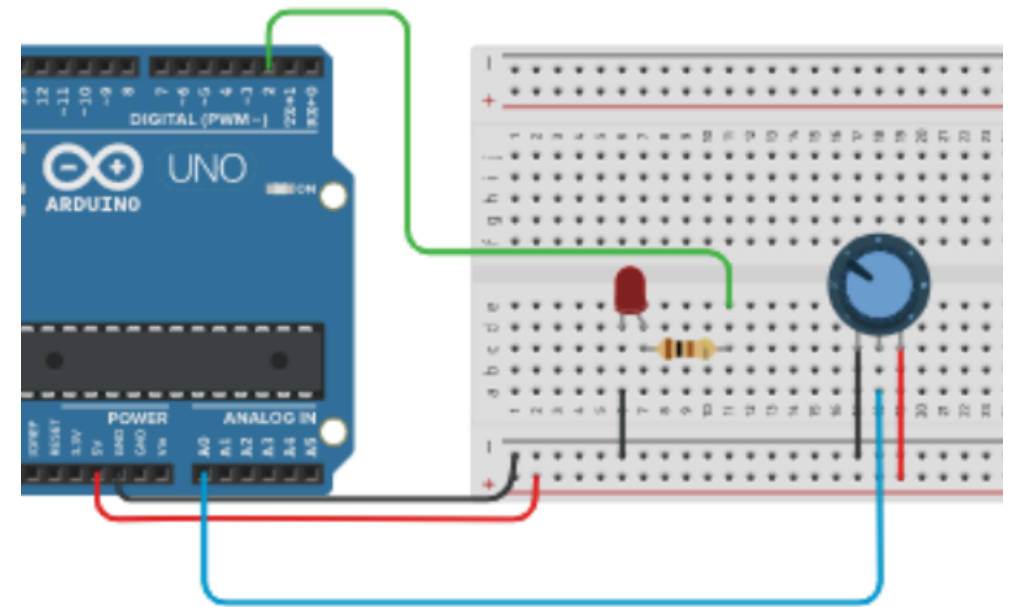


Ejemplo 5: Control intensidad LED

Realizar un código, que a partir de la lectura de voltaje de un potenciómetro se controle la intensidad de un LED mediante una salida analógica

Pasos:

1. Crear variables
2. Leer valor Pin analógico y conversión.
3. Escribir el valor de voltaje en un Pin de salida analógica. Cuidado, las salidas son de 8 bit !!!
4. Mostrar el valor de voltaje por puerto serie.



Ejemplo 5: Control intensidad LED

Realizar un código, que a partir de la lectura de voltaje de un potenciómetro se controle la intensidad de un LED mediante una salida analógica

Pasos:

1. Crear variables
2. Leer valor Pin analógico y conversión.
3. Escribir el valor de voltaje en un Pin de salida analógica. Cuidado, las salidas son de 8 bit !!!
4. Mostrar el valor de voltaje por puerto serie.

```
int valor;
float voltaje;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  valor = analogRead(A0);
  voltaje = valor*5/1023.0;
  analogWrite(3,valor/1023.0*255); //8 bits
  Serial.println(voltaje);
  delay(500);
}
```

Ejemplo 6: Bucles For

- Crear un código que cree una señal triangular y la muestre por monitor (plotter).

Pasos:

- Utilizar un bucle *for* que muestre los datos de subida por monitor.
- Utilizar un bucle *for* que muestre los datos de bajada por monitor.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  for(int i=0;i<10;i=i+1) {
    Serial.println(i);
  }
}
```


Ejemplo 6: Bucles For

- Crear un código que cree una señal triangular y la muestre por monitor (plotter).

Pasos:

- Utilizar un bucle *for* que muestre los datos de subida por monitor.
- Utilizar un bucle *for* que muestre los datos de bajada por monitor.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  for(int i=0;i<10;i=i+1) {
    Serial.println(i);
  }
  for(int i=10;i>0;i=i-1) {
    Serial.println(i);
  }
}
```

Ejemplo 7: Vector de datos

- Crear un código que cree un vector con los valores analógicos medidos y después muestre su media aritmética.
- Pasos:
- Inicializar un vector antes del `setup()` (`int vec[10]`)
- En el void `loop()` crear un `for` que cada segundo guarde un dato en una posición del vector creado.
- Hacer otro `for` que recorra todos los datos del vector, calcule la media aritmética y los envíe por el puerto serie.

```
int vec[10];
float suma;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  for(int i = 0; i < 10; i = i+1){
    vec[i] = XXXXXXXXXX;
    delay(100);
  }

  suma = 0;
  for(int i = 0; i < 10; i = i+1){
    suma = XXXXXXXXXX;
  }

  Serial.println(suma/10.0);
}
```

Ejemplo 7: Vector de datos

- Crear un código que cree un vector con los valores analógicos medidos y después muestre su media aritmética.
- Pasos:
- Inicializar un vector antes del `setup()` (*int vec[10]*)
- En el void `loop()` crear un *for* que cada segundo guarde un dato en una posición del vector creado.
- Hacer otro *for* que recorra todos los datos del vector, calcule la media aritmética y los envíe por el puerto serie.

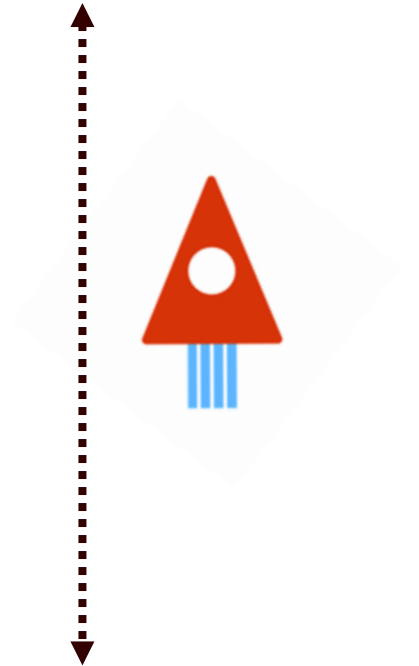
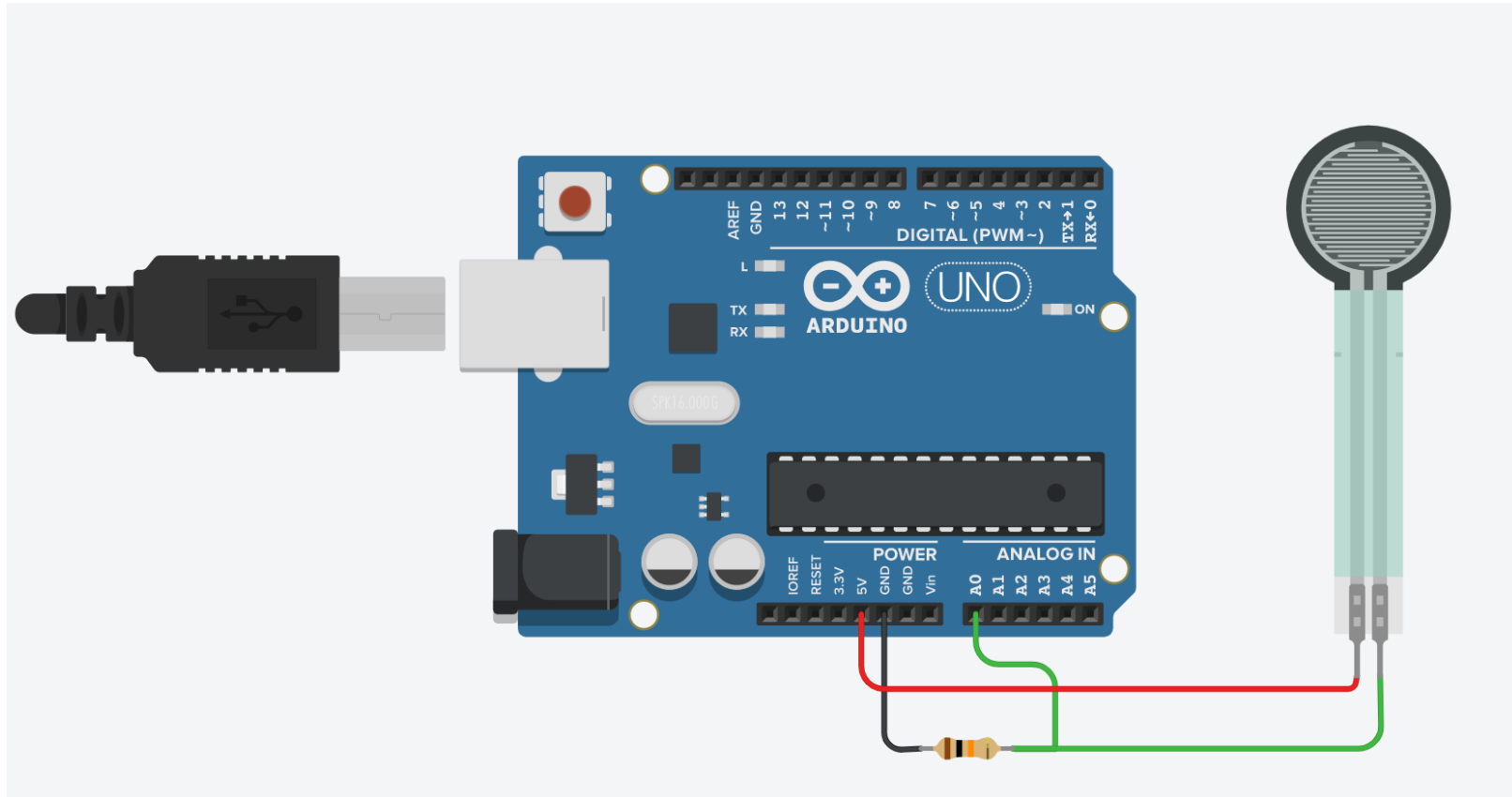
```
int vec[10];
float suma;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  for(int i = 0; i < 10; i = i+1){
    vec[i] = analogRead(A0)*5/1023.0;
    delay(100);
  }

  suma = 0;
  for(int i = 0; i < 10; i = i+1){
    suma = suma + vec[i];
  }

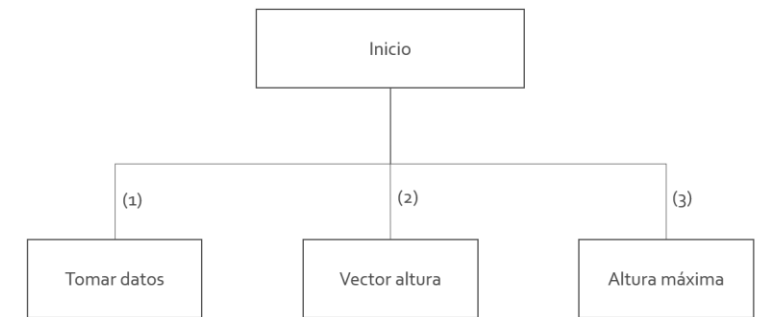
  Serial.println(suma/10.0);
}
```

Programa ejemplo telemetría



Programa ejemplo telemetría

- Leer el puerto serie y guardar su valor en una variable *char*.
- Si el valor es '1':
 - Llenar un vector de 100 datos con la lectura de una de las entradas analógicas, con una frecuencia de 5 Hz.
 - Antes de empezar el llenado, enviar por puerto serie "Tomando datos...".
 - Al finalizar, enviar "Todos los datos guardados".
- Si el valor es '2'
 - Enviar por puerto serie los valores del vector guardado.
 - Antes de enviar los datos, enviar por puerto serie "Vector de datos [m]"
- Si el valor es '3'
 - Enviar el valor máximo de altura (Hacer un bucle que recorra todo el vector comparando una variable con el valor del vector, si el valor del vector es mayor, se actualiza la variable).



Programa ejemplo telemetría

```
int i, h[40];
char valor;

void setup()
{
  Serial.begin(9600);
}

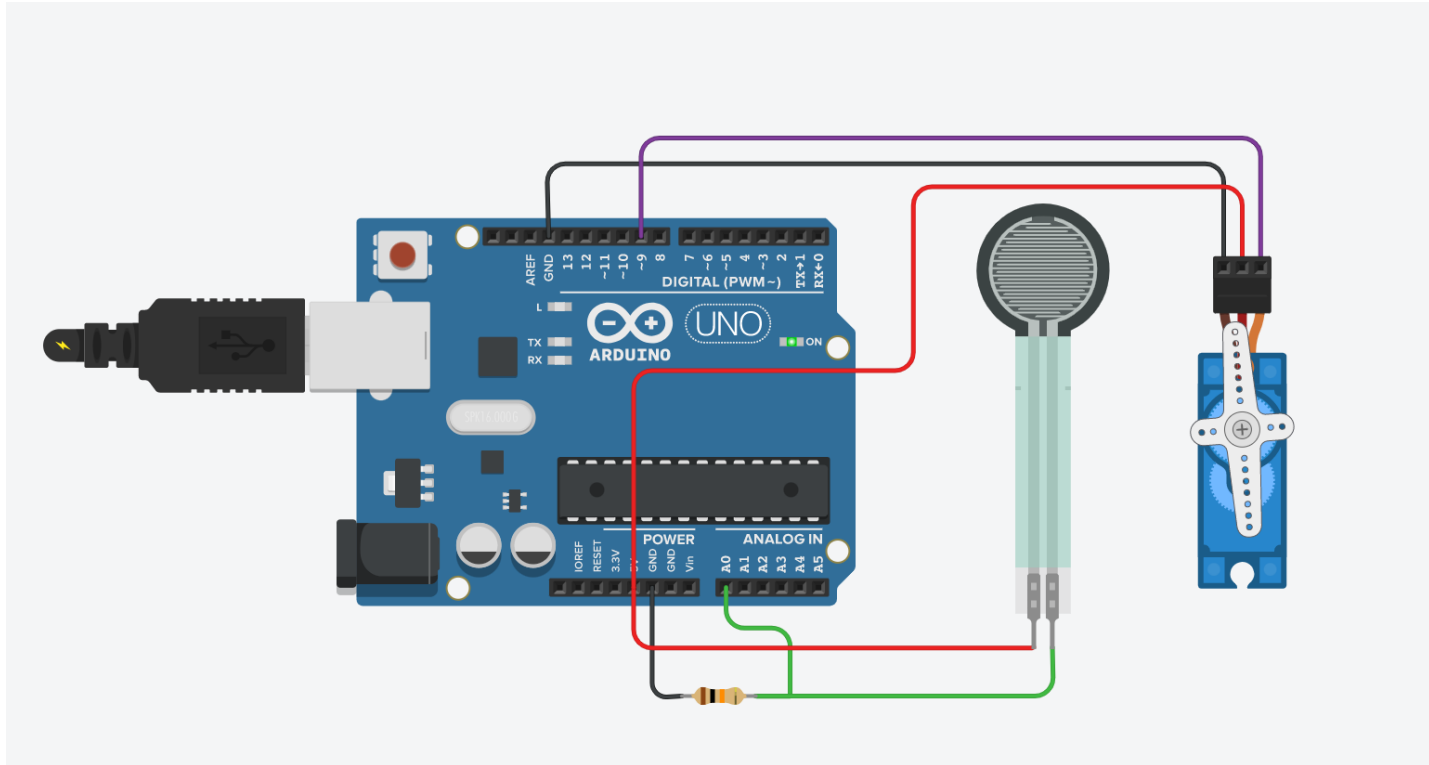
void loop()
{
  if(Serial.available()){
    valor=Serial.read();

    if(valor=='1'){
      Serial.println("Tomando datos...");
      for(i=0;i<40;i=i+1){
        h[i]=analogRead(A0);
        delay(200);
      }
      Serial.println("Todos los datos tomados");
    }
  }
}
```

```
if(valor=='2'){
  for(i=0;i<40;i=i+1){
    Serial.println(h[i]);
  }
}

if(valor=='3'){
  int max=0;
  for(i=0;i<40;i=i+1){
    if(h[i]>max) max=h[i];
  }
  Serial.println("Altura max");
  Serial.println(max);
}
}
```

Programa ejemplo telemetría



¿Paracaídas?

**WATER
ROCKETS**