



Co-funded by the
Creative Europe Programme
of the European Union

Project 2020-1-TR01- KA201-094533



The Key To Global Life,
Digital Change Of Nature



Duración total: 2 o 3 horas



Edad del estudiante: 14-18 años



- Área de aplicación:
- Cambio climático
- física
- electrónica



Palabras clave: Temperatura,
análisis de datos, clima, Ardui-
no, tecnología.



G1 - Sistema de Monitoreo de Temperatura con Arduino UNO



- Módulo
- Calentamiento global

G1 - Versión en español

Materiales:

Papel (A3)

(Lápices de colores

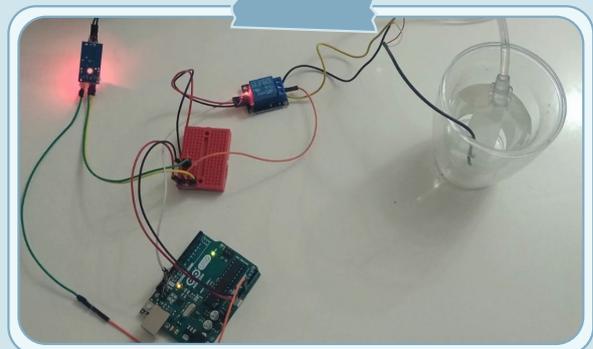
Computadora portátil

Arduino (www.arduino.cc)

Tablero de circuitos

Sensor de temperatura (LM35 o alternativo)

Cables de puente



- Notas:
- Tamaño de cada grupo: 3-4 estudiantes.
- Fase 1: hazlo primero en un entorno virtual.
- o Utilizar Tinkercad (circuitos):
- o Dibuje un boceto ordenado y detallado de la configuración: incluya el sensor de temperatura, Arduino y la placa de pruebas.
- o Conecte todo junto, usando diferentes colores de "cable" y use el color correcto de cable para los polos positivo y negativo.
- o Realizar el programa sobre el circuito en Tinkercad y probarlo primero online.
- Fase 2: Hazlo realidad:
- o Encuentra todos los componentes y



@digitalchangeon

Introducción

Construiremos una instalación de bricolaje portátil y asequible que monitoreará la temperatura y la intensidad de la luz. Usamos un Arduino UNO para recopilar datos y lo vinculamos a una computadora para mostrar los datos en Excel. Para realizar esta actividad el docente necesita tener conocimientos básicos en la materia de Arduino, Arduino IDE y uso básico de Excel.

El tiempo y el clima son de gran importancia en la sociedad actual. El aumento de la potencia informática nos permite ejecutar modelos meteorológicos a resoluciones cada vez más altas, pero esto crea la necesidad de más datos meteorológicos locales. Imagine un país donde cada escuela monitorea el clima, podemos obtener datos a nivel del suelo muy precisos (Imagen 1).

Las medidas no sólo son importantes, sino que también es necesario visualizar lo que se mide.

Construiremos una instalación de bricolaje portátil y asequible que monitoreará la temperatura y la intensidad de la luz. Usamos un Arduino para recopilar datos y lo vinculamos a una computadora para mostrar los datos en Excel.

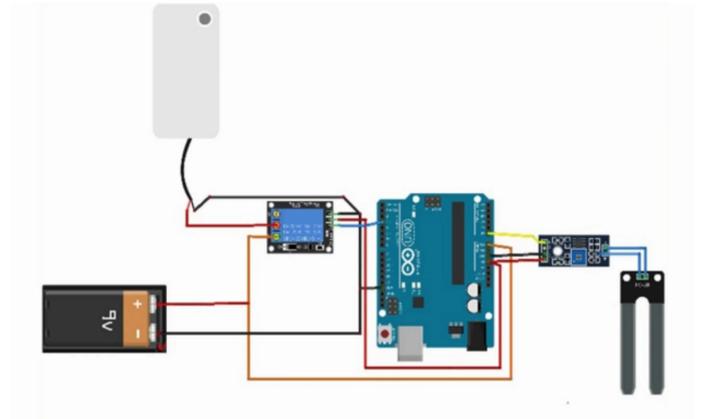


Imagen 1. Sistema de Monitoreo de Tempurtare

Consideraciones

Tamaño de cada grupo: 3-4 estudiantes.

Fase 1: hazlo primero en un entorno virtual

Utilice Tinkercad (circuitos):

Dibuje un boceto ordenado y detallado de la configuración: incluya el sensor de temperatura, Arduino y la placa de pruebas.

Conecte todo junto, usando diferentes colores de "cable" y use el color correcto de cable para los polos positivo y negativo.

Haz el programa en el circuito en Tinkercad y pruébalo primero online.

Fase 2: Hazlo realidad:

Encuentra todos los componentes y conecta todo.

¡No conectes el Arduino a la PC todavía! El profesor lo comprueba primero.

Objetivo de la actividad

- Facilite la medición de la temperatura a intervalos regulares
- Utilice una herramienta de análisis como Excel que sea ampliamente conocida para recopilar y visualizar datos.
- Aprenda a trabajar con sensores en general para recopilar datos:
- Comprender la relación entre el voltaje medido y la cantidad física a medir.
- Conozca el concepto de calibrar un sensor, aprenda a trabajar con una ficha técnica
- Aprende a programar en C++
- Familiarízate con las funciones lineales.



Imagen 2. Objetivo de la actividad

Proceso de actividad

Antes de la ac-



Imagen 3. Mezclar con una cuchara.

- Explique la tarea: antecedentes, objetivo, plazo de cada parte.
- Divida la clase en grupos de 2 a 4 estudiantes, cada grupo en su propia mesa. Cada grupo tiene una computadora portátil, papel y lápices.

Empecemos

1 Diseño de circuitos (30 minutos)

- Dibuja el circuito en una página.
- En su lugar, puedes utilizar Tinkercad Circuits para dibujarlo en línea. Lo que necesitas es (Imagen 3 y 4).
- un arduino
- una placa de pruebas
- el sensor de temperatura (LM35 o TMP36)



Imagen 4. Materiales

2 ¡Conéctalo!

- Coloque el sensor de temperatura en la placa, asegúrese de que esté colocado de la manera correcta para que las clavijas no sufran un cortocircuito.
- El sensor necesita voltaje para funcionar. Puede utilizar el pin de 5V del Arduino. Consulte la figura para conocer la disposición correcta de los pines. Conecte el pin 1 al pin de 5V del Arduino. Conecte el GND al GND del Arduino.
- El Arduino tiene que “escuchar” las lecturas del sensor. Esta es una entrada analógica, debes usar uno de los pines de entrada analógica del Arduino. Conecte el pin central del sensor al A0.

3 Programming: display the output of the sensor

Comenzamos leyendo lo que tu Arduino recibe del sensor (la ENTRADA) y lo mostramos en el Monitor Serial. Esto se hace programando en C++. Comenzaremos con un programa simple que lee la lectura del sensor cada segundo y la imprime en la consola de la computadora (Imagen 5).



- Para leer el voltaje en el pin A0, use `analogRead(A0)`.
- Almacena el voltaje en una variable llamada "valor". En este lenguaje de programación (C++) siempre tienes que decir qué tipo de información quieres almacenar en una variable cuando creas la variable. Crear la variable se llama "declarar" y eso sucede en la línea 1 del programa. En este caso, la variable contendrá un número entero, llamado "entero". De ahí el valor `int` en la parte superior del código.

```

1 int value;
2
3 void setup() {
4     Serial.begin(9600);
5 }
6
7 void loop() {
8     value = analogRead(A0);
9     Serial.println(value);
10    delay(1000);
11 }

```

Imagen 5. Códigos C++



Tabla 1. Otros tipos de información en C++

| | |
|---------------------|--|
| <code>int</code> | Número entero (0, 1, 2, -3, 19839, -78,...) |
| <code>float</code> | Número decimal (0,13, 713,24, -3,0,...) |
| <code>char</code> | Un solo carácter ('a', 'B', 'c', 'D', '0', '*', 'µ',...) |
| <code>String</code> | texto ("¡Hola mundo!", "esto es una prueba", "123",...) |

4

Comunicación entre Arduino y PC.



- Como recordará, "carga" un programa en Arduino y el programa se ejecuta en Arduino, no en la computadora. Es el Arduino el que lee el valor del sensor, pero el ordenador tiene que mostrar este valor en la pantalla. El Arduino debe enviar esta información primero a la computadora.
- Para configurar una conexión entre Arduino y la computadora, use `Serial.begin(9600)` en el bloque de configuración (porque solo necesita configurar la conexión una vez, al principio). El número entre paréntesis () le dice a la computadora cuántos bits enviará por segundo. Esto se llama "velocidad en baudios". Cuanto mayor sea este número, más datos se enviarán por segundo, pero también mayor será la posibilidad de que se produzcan errores durante la comunicación (que un 0 se convierta accidentalmente en un 1 y viceversa). Para algunas aplicaciones es importante alcanzar una alta velocidad, por ejemplo en la impresión 3D. 9600 es más que suficiente para las aplicaciones que utilizamos.
- Después de que Arduino haya establecido comunicación con la computadora y haya leído el valor del pin central y lo haya almacenado en una variable, es hora de enviar este valor a la computadora.
- `Serial.println(valor)` hace que se imprima una línea en la computadora (de ahí `println`, traducido como "imprimir línea"). Ponga entre paréntesis lo que se debe imprimir.



- La salida se puede encontrar en el 'Monitor serie'. En Tinkercad puede encontrar esto haciendo clic en "Monitor de serie" en la parte inferior de la pantalla Código (Imagen 6).
- En Arduino IDE puedes abrirlo haciendo clic en "Monitor serie" en el menú "Herramientas". O haga clic en el icono (Imagen 7).



Imagen 6. Monitor serie en Tinkercad

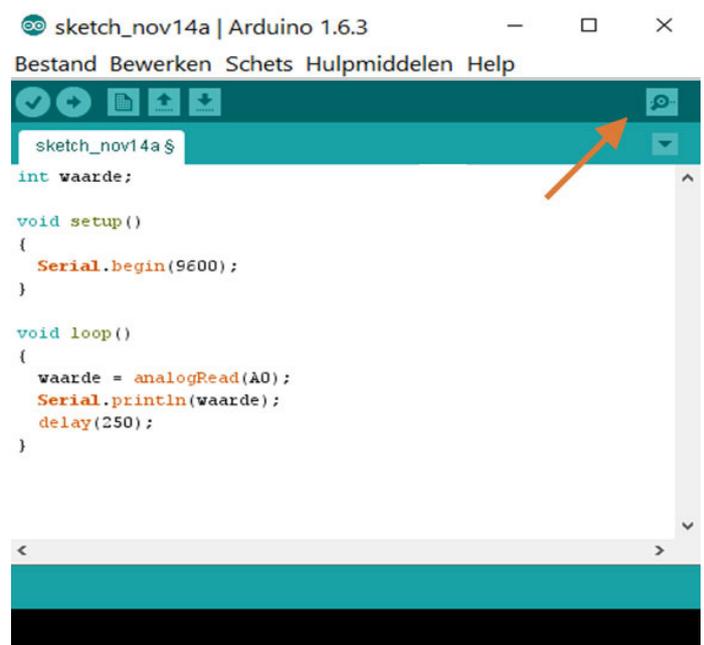


Imagen 7. Monitor serie en Arduino IDE

5 Programación: calibrar el sensor

- Recibes un cierto valor de tu sensor, pero ¿qué significa ese valor exactamente? Por eso es importante que calibres tu sensor, para que puedas interpretar correctamente un determinado valor. En este caso: ¿qué significa el valor 523? ¿Es "cálido" o "frío"? Esta vez, tenemos que buscar en la hoja de datos del sensor LM35/TMP36 para encontrar las especificaciones. Si los encontramos, sabemos cómo convertir la lectura a la temperatura correcta.



- Agregue la palabra clave



- Busca en internet cuales son las especificaciones del sensor de temperatura. Anota en la siguiente tabla lo que has encontrado.

| Condición | Valor medido teórico |
|------------------------------|----------------------|
| Temperatura mínima | |
| temperatura máxima | |
| Pendiente (mV/grado Celsius) | |
| Exactitud | |
| Vaciado a 150°C | |
| Salida a 0°C | |



Ahora tenemos que encontrar una fórmula para calcular nuestra temperatura. Pero, ¿cómo funciona `analogRead()`?

Arduino siempre traduce el voltaje medido en el pin analógico (0-5V) a un número (0-1023).

0V se traduce en 0

5V se traduce en 1023

Entonces tienes que dividir por 1023 y multiplicar por 5 nuevamente para pasar del valor al voltaje.

La siguiente parte depende del sensor que utilices, ¡así que ten cuidado de utilizar las especificaciones correctas! Nuestro ejemplo a continuación es válido sólo para LM35:

Sabemos por las especificaciones de la hoja de datos que

0°C se traduce en 0V

150°C se traduce en 150mV = 0,150V

5000°C se traduce en 5000mV = 5V

Entonces cada 10mV (0.01V) es 1°C.

Tienes que multiplicar el voltaje medido por 100 para obtener los °C.

La fórmula para calcular la temperatura a partir del valor resultante de `analogRead()`

Temperatura = $(5,0 * \text{valor} * 100,0) / 1023,0$



Nuestro programa para medir y calcular la temperatura con un sensor LM35 (Imagen 8)



Tenga cuidado, los números decimales se muestran con un punto (.) y no

```

1 int value = 0;
2 float temperature = 0;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   value=analogRead(A0);
10  Serial.print("our sensor reads: ");
11  Serial.println(value);
12  temperature = 5.0 * value * 100.0 / 1023.0;
13  Serial.print("this means in degrees Celsius: ");
14  Serial.println(temperature);
15  delay(1000);
16 }
```

Imagen 8. Calcular la temperatura



Para los estudiantes inteligentes que terminaron más rápido:
Cálculalo para otro tipo de sensor como TMP36, TMP37 o los muy utilizados DHT11 o DHT22.

6 Capturar los datos en Excel.

Ya podemos imprimir los valores a través del monitor Serial. Ahora veremos cómo transmitir estos datos a Excel. Esto podría resultar útil.

No hay cambios en la entrada ni en su circuito. Todo lo que necesitamos hacer es habilitar y configurar un complemento en Excel. Esto captura los datos que se envían al monitor serie y muestra estos valores en una cantidad de celdas configuradas para este propósito.

El complemento que necesitamos se llama 'Microsoft Data Streamer for Excel' y está presente de forma predeterminada en las últimas versiones de Microsoft Excel. Aún tienes que habilitarlo.

Haga clic en:

'Archivo' >> 'Opciones' >> "Complementos"

haga clic en la flecha junto a "Administrar: complementos de Excel" >> 'Complementos COM' >> 'Iniciar'

Marque la casilla junto a 'Microsoft Data Streamer para Excel'
DE ACUERDO'

Ahora aparecerá una nueva pestaña en la cinta, en el extremo derecho, llamada "Data Streamer". Ahora abra la pestaña "Transmisor de datos". Haga clic en "Conectar dispositivo" y elija "Arduino Uno (COMX)".

Se crea un nuevo libro de trabajo. Para transmitir datos, haga clic en "Iniciar datos". Cada segundo tiempo y temperatura ahora se imprimen aquí. La hora se agrega automáticamente mediante Excel. En la hoja de trabajo 'Configuración' puede ajustar, entre otras cosas, cuántas filas de datos deben ser visibles (máximo 500). Ahora establezca el número de filas de datos en 50.

Modifica el programa de tal forma que solo imprima los valores, no el texto. Ahora imprima ambos valores en la misma línea, separados por una coma.

```
Serie.print(valor);  
Serie.print(",");  
Serial.println(temperatura);
```

7 Graficos

Haz dos gráficas lineales, una para el valor medido y otra para la temperatura. Si todo ha ido bien, estos gráficos deberían actualizarse en directo.

8 Ahorrar

También puede guardar datos para su uso posterior. Para ello haga clic en el botón "registrar datos". Después de detener la grabación, se le preguntará dónde desea guardar el archivo. El formato de este archivo es ".csv", abreviatura de "valores separados por comas". Puede leer este formato en muchos programas relacionados con datos posteriormente,

9 Ahora modifíquelo, amplíelo



- Utilice un segundo sensor, p. un LDR, sensor de humedad del suelo,... Agregue los valores del sensor que lea al Excel. Para el sensor de humedad del suelo, calcule la humedad relativa.
- Pon todo en una visualización gráfica.

Evaluación

Evaluación

El diseño de los estudiantes se puede exhibir dentro de la escuela. Se pueden crear diferentes productos diversificando los materiales de desecho utilizados.

| Objetivos | Debe ser mejorada (1) | Debe ser mejorado (1) | Bien (3) | Muy bien (4) |
|---|-----------------------|-----------------------|----------|--------------|
| (Re)formular un problema para que pueda resolverse mediante una computadora u otra herramienta. | (....) | (....) | (....) | (....) |
| Participación activa | (....) | (....) | (....) | (....) |
| Crear un conjunto de instrucciones para alcanzar un objetivo desde un punto de partida (= algoritmo) se logra de forma independiente. | (....) | (....) | (....) | (....) |
| La selección e implementación de información útil de una fuente específica se logra de forma independiente. | (....) | (....) | (....) | (....) |

Links

Mundo colorido. (2018). Cómo hacer fácilmente un pingüino, hielo y un iglú por una pequeña cantidad de dinero / bricolaje. <https://www.youtube.com/watch?v=Es-rCelq6YU>

Cristofari, R., Liu, X., Bonadonna, F., Cherel, Y., Pistorius, P., Le Maho, Y., . . . Trucchi, E. (2018). Cambios de distribución del pingüino rey impulsados por el clima en un ecosistema fragmentado. *Naturaleza Cambio Climático*, 8(3), 245-251.

Amigos de la ciencia. (2020). Corrientes oceánicas: modelando la "cinta transportadora global" en su cocina. Obtenido el 10 de octubre de 2022 de https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p012/ocean-sciences/ocean-currents-modeling-global-conveyor-belt

- <https://www.sciencebuddies.org/stem-activities?s=global%20warming>
- <https://www.sciencebuddies.org/stem-activities/polar-ice-caps-melting>
- https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p015/ocean-sciences/will-ice-melting-at-poles-cause-sea-levels-to-rise
- <https://www.tinkercad.com/things/c3BkCJdQxel>
- <https://www.tinkercad.com/things/9UeZJTri0zD>
- <https://www.youtube.com/watch?v=Gkw45JaEQio>
- <https://www.youtube.com/watch?v=ztQYbRwBboU>
- <https://science.howstuffworks.com/environmental/earth/oceanography/ocean-current.htm>
- <https://web.ics.purdue.edu/~braile/edumod/convect/convect.htm>