**Laboratorio 2: Estrusturas de Programacion en Arduino**

**Alumno: Alvaro Palomino Nolasco**

**Objetivos:**

Al finalizar el laboratorio el estudiante será capaz de:

* Definir y programar variables en Arduino IDE
* Trabajar con definiciones en Arduino IDE.
* Definir y programar Estructuras Decisivas en Arduino IDE.
* Definir y programar Estructuras Repetitivas en Arduino IDE.

**Seguridad:**

USANDO SIMULADOR:

* Registrarse en la web de simulación con el correo de Tecsup y no olvidarse de la clave.
* Prestar atención a las indicaciones del docente.

USANDO COMPONENTES REALES:

* Utilizar pulsera antiestática para evitar la descarga estática en los circuitos, sino se tiene evitar trabajar con chompa o abrigo de lana ya que dicho material genera carga eléctrica.
* Realizar el armado del circuito propuesto **sin tener conectado ninguna fuente de energía**
* Verificar el armado del circuito antes de energizar para evitar averias.
* No realizar ningún cambio en el circuito si se tiene energizado.
* Prestar atención a las indicaciones del docente.

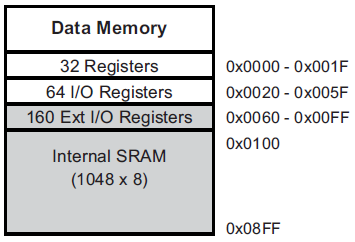
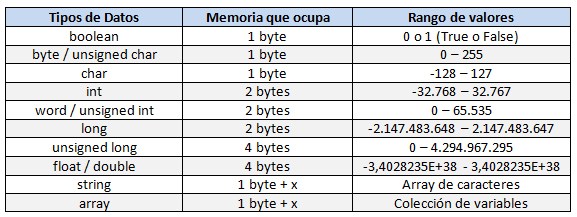
**Equipos y Materiales:**

* Una computadora con:
* Windows 7 o superior
* Conexión a Internet
* Arduino y elementos adicionales.

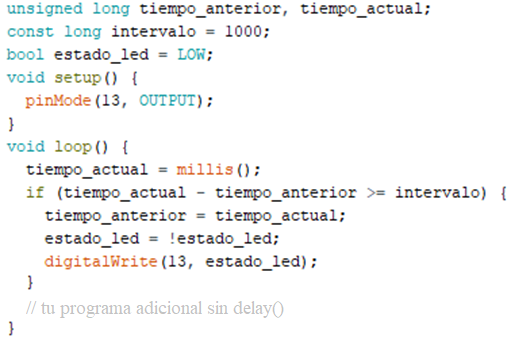
**Procedimiento:**

**EJERCICIO 1: Variable en Arduino IDE**

*En Arduino IDE las variables tiene diferentes tamaños de consumo de memoria de datos.*

La forma de declarar una variable es asignándole un tipo de dato y un nombre con dígitos ALFANUMERICOS y el SUBGUION, aquí tenemos un ejemplo:



1. Realice un programa en Arduino IDE que tenga dos variables nombradas ***numero1*** y ***numero2***de tipo***int***(Entero),envié el resultado de las 4 operaciones aritméticas básicas por el puerto serial.

Comparta el link de la actividad con el docente:

<https://www.tinkercad.com/things/awW7rOulb0d-dazzling-robo-jaiks/editel?tenant=circuits>

|  |  |
| --- | --- |
| Coloque el programa utilizado en la actividad: |  |
| Coloque el circuito electrónico utilizado en la actividad: |  |

*Responda:*

1. Explique ¿Qué diferencia se tiene en una variable de tipo ***int*** y ***long*** en Arduino?

|  |
| --- |
| En Arduino, las variables de tipo int ocupan 2 bytes y pueden representar valores en el rango de -32,768 a 32,767, mientras que las variables de tipo long ocupan 4 bytes y pueden representar valores en el rango de -2,147,483,648 a 2,147,483,647. La diferencia principal radica en el rango de valores que pueden almacenar, siendo int más adecuado para valores más pequeños y long para valores más grandes. |

1. Si se opera aritméticamente dos variables del mismo tipo ¿En qué características debemos fijarnos?

|  |
| --- |
| Cuando se operan aritméticamente dos variables del mismo tipo en Arduino, es importante considerar el posible desbordamiento de variables si el resultado excede el rango de valores que puede representar el tipo de variable. También debemos tener en cuenta el uso de memoria, especialmente al trabajar con variables long, que ocupan más espacio. Además, la precisión de los cálculos puede verse afectada, especialmente al realizar divisiones entre enteros, lo que puede resultar en la pérdida de decimales. Por lo tanto, debemos ser conscientes de estas características para asegurarnos de que nuestro código funcione correctamente y produzca resultados precisos. |

**EJERCICIO 2: Definiciones en Arduino IDE**

*En Arduino IDE las definiciones sirve para hacer el programa más legible para el programador, el compilar se encargará de reemplazarlo antes de compilar.*

Se puede definir cualquier objeto, dicha DEFINICION estaría denominado como un sobrenombre, se define con la palabra reservada *#define*. Las reglas de nombre son iguales a los nombres de variables:





1. Realice un contador con 2 definiciones para números con nombres de ***valor\_min*** y ***valor\_max*** de modo que se tenga un contador ascendente cada medio segundo entre ***valor\_min*** y ***valor\_max***, si la cuenta llega al ***valor\_max*** se reinicia a ***valor\_min***, enviar la cuenta por el puerto serial:

Comparta el link de la actividad con el docente:

<https://www.tinkercad.com/things/k0qlxwcBvnE-grand-wolt/editel?tenant=circuits>

|  |  |
| --- | --- |
| Coloque el programa utilizado en la actividad: |  |
| Coloque el circuito electrónico utilizado en la actividad: |  |

*Responda:*

1. Explique ¿Qué diferencia se tiene en una variable y una definición en Arduino?

|  |
| --- |
| En Arduino, una variable es un espacio en la memoria que puede contener diferentes valores durante la ejecución del programa y se declara con un tipo de datos y un nombre. Una definición, por otro lado, es una directiva del preprocesador que asigna un nombre a un valor constante o una expresión para mejorar la legibilidad del código y evitar repeticiones. La diferencia clave radica en que una variable puede cambiar su valor durante la ejecución del programa, mientras que una definición establece un valor constante que se utiliza en todo el código. |

1. En un programa de Arduino ¿Cuántas definiciones podemos tener?

|  |
| --- |
| En un programa Arduino, puedes tener tantas definiciones como necesites. |

¿Porque?

|  |
| --- |
| Porque no hay un límite específico en la cantidad de definiciones que puedes tener, pero es recomendable utilizarlas de manera moderada para mantener la legibilidad del código. Es importante organizar las definiciones de manera coherente y mantener una estructura clara en el código para facilitar su comprensión y mantenimiento. |

**EJERCICIO 3: Estructuras Decisivas en Arduino**

*En Arduino IDE las estructuras decisivas nos permiten realizar la toma de decisión de acuerdo a condiciones.*

**IF-ELSE / SWITCH-CASE:**

La estructura if-else o switch-case nos permiten analizar diferentes condiciones y bifurcar el flujo del programa de acuerdo a su estado:

|  |  |
| --- | --- |
|  |  |

1. Realice un programa, si el Arduino recibe por el puerto serial la Letra ‘A’ responder la palabra ‘CORRECTO’, de lo contrario enviar ‘INCORRECTO’

Comparta el link de la actividad con el docente:

<https://www.tinkercad.com/things/kTN82mzeE3E-bodacious-allis-elzing/editel?tenant=circuits>

|  |  |
| --- | --- |
| Coloque el programa utilizado en la actividad: |  |
| Coloque el circuito electrónico utilizado en la actividad: |  |

*Responda:*

1. Explique ¿Cuál de las 2 estructuras decisivas anteriores le parece más util?

|  |
| --- |
| La estructura if-else es más versátil y útil para evaluar condiciones complejas y manejar múltiples casos independientes. Por otro lado, la estructura switch-case es más adecuada cuando necesitas evaluar una sola expresión contra múltiples valores posibles, ofreciendo un código más compacto y legible en tales casos específicos. La elección entre una u otra depende del contexto y las necesidades del problema a resolver. |

1. Podría mencionarnos ¿En qué casos usaría ***if-else*** y en qué casos usaría ***switch-case***?

|  |
| --- |
| Usa la estructura if-else cuando necesites evaluar condiciones complejas o múltiples condiciones independientes. Emplea switch-case cuando tengas una sola expresión que deba compararse con múltiples valores posibles y prefieras un código más compacto y legible. |

**EJERCICIO 4: Estructuras Repetitivas en Arduino**

*En Arduino IDE las estructuras repetitivas nos permiten realizar la repetición de acuerdo a condiciones establecidas.*

**FOR / WHILE:**

La estructura ***for*** se utiliza para realizar repeticiones contadas o finitas y ***while*** se utiliza para repeticiones no contadas o posiblemente infinitas, todo de acuerdo a las condiciones colocadas:

|  |  |
| --- | --- |
|  |  |

1. Realice un programa, utilizando la estructura ***for*** realice un contador ascendente cada medio segundo del 5 al 15 y repetir dicha cuenta:

Comparta el link de la actividad con el docente:

<https://www.tinkercad.com/things/c3Me3tJIkcs-super-wolt/editel?tenant=circuits>

|  |  |
| --- | --- |
| Coloque el programa utilizado en la actividad: |  |
| Coloque el circuito electrónico utilizado en la actividad: |  |

*Responda:*

1. Explique ¿Explique el funcionamiento de la estructura ***for***?

|  |
| --- |
| La estructura for en Arduino permite realizar bucles controlados con tres partes: inicialización, condición y actualización. Se ejecuta la inicialización una vez al inicio, luego se evalúa la condición antes de cada iteración; si es verdadera, se ejecuta el cuerpo del bucle y se actualiza la variable de control al final de cada iteración. Este proceso se repite hasta que la condición se vuelva falsa. Es una forma compacta y conveniente de realizar repeticiones controladas en el código. |

1. Explique ¿Explique el funcionamiento de la estructura ***while***?

|  |
| --- |
| La estructura while en Arduino se utiliza para realizar bucles basados en una condición. Funciona evaluando la condición antes de cada iteración y ejecutando el bloque de código dentro del bucle mientras la condición sea verdadera. Es importante evitar bucles infinitos asegurándose de que la condición eventualmente se vuelva falsa. El while es útil cuando no se sabe cuántas veces se debe repetir un bloque de código, pero se quiere que se ejecute mientras se cumpla una condición específica. |

**TAREA 1: Simule en Arduino**

1. Escriba un programa en Arduino que reciba por el puerto serial su apellido y el Arduino responda “Presente”:

Comparta el proyecto con el Docente y coloque el link del proyecto:

<https://www.tinkercad.com/things/187YMIOfHei-daring-densor-blad/editel?tenant=circuits>

Describa el programa que realizo ¿Que funciones uso?

El programa utiliza funciones de comunicación serial para recibir un apellido del usuario. Luego, convierte el apellido a minúsculas para una comparación insensible a mayúsculas y minúsculas y verifica si coincide con "quispe". Si coincide, responde con "Presente" a través del puerto serial. Las funciones principales utilizadas son Serial.begin(), Serial.available(), Serial.readStringUntil(), toLowerCase(), equals() y Serial.println().

**TAREA 2: Video de Simulación**

1. Realice un video de máximo 1 minuto y súbalo al [www.youtube.com](http://www.youtube.com) o DRIVE y coloque el link:

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

Imagen que contiene computadora, circuito

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**TAREA 3: INVESTIGACIÓN**

Averigüe sobre ¿Qué es señal digital y señal analógica?

Una señal digital es aquella que toma valores discretos, típicamente representados como "0" o "1", y se utiliza en sistemas electrónicos digitales como computadoras y teléfonos móviles. Por otro lado, una señal analógica es continua y puede tomar cualquier valor dentro de un rango, utilizada en aplicaciones como audio y video. Las señales digitales son más robustas frente al ruido, mientras que las analógicas son más susceptibles a interferencias. Cada tipo de señal tiene sus propias ventajas y se aplica en función de las necesidades específicas del sistema.

**OBSERVACIONES**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CONCLUSIONES.**

* exploramos las estructuras if-else y while, que son fundamentales para controlar el flujo de ejecución en programas Arduino. Estas estructuras nos permiten tomar decisiones basadas en condiciones y realizar repeticiones basadas en condiciones específicas.
* discutimos las diferencias entre las señales digitales y analógicas. Las señales digitales son discretas y se utilizan en sistemas digitales, mientras que las señales analógicas son continuas y se utilizan en aplicaciones donde se requiere una representación continua de la información.
* Hemos utilizado la comunicación serial para interactuar con el Arduino a través del puerto serial. Esto nos permite enviar y recibir datos entre el Arduino y otros dispositivos, como una computadora, utilizando el monitor serial.
* Hemos desarrollado programas simples en Arduino utilizando las estructuras de control y la comunicación serial. Estos programas ilustran cómo podemos utilizar las funciones disponibles en Arduino para realizar diversas tareas, como tomar decisiones basadas en condiciones y responder a eventos externos.