

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Organización Computacional  
Ing. Juan Carlos Maeda Juárez  
Aux: Javier Gutierrez



## **PROYECTO FINAL**

### **Lector de braile**

## **OBJETIVOS**

### **Objetivo General**

1. Poner en práctica todos los conocimientos adquiridos en el laboratorio.

### **Objetivos Específicos**

1. Mejorar el uso de los distintos tipos de Flip Flops y contadores.
2. Implementar una transmisión serial a través de los puertos de una PC.
3. Desarrollar una estructura mecánica funcional.
4. Aplicar e Implementar conocimientos de registros.
5. Implementar software para el control de puertos.
6. Aplicar e Implementar conocimientos de memorias.
7. Aprender conocimientos de lógica secuencial y simplificación de estados.
8. Aprender el funcionamiento y uso de motores paso a paso (Stepper).
9. Aprender el uso de distintos tipos de sensores.
10. Integrar circuitos eléctricos y producir movimientos mecánicos.
11. Aplicar todos los conocimientos aprendidos en el curso
12. Aplicar los conocimientos adquiridos de Arduino.
13. Aprender la aplicación de una memoria RAM.

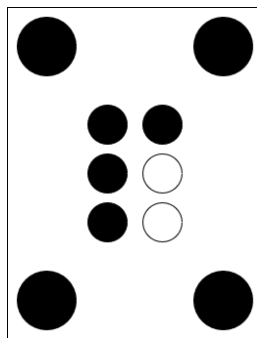
## DESCRIPCIÓN

Prociegos y Sordos de Guatemala con el fin de lograr una verdadera inclusión ha creado muchos libros educativos para personas ciegas que abarcan grados desde primaria hasta diversificado, el problema es que con el pasar del tiempo muchas de las páginas de los libros se han apachado y los lectores ya no pueden entender las letras, por lo que usted con los conocimientos adquiridos en el curso de Organización Computacional deberá crear un prototipo de verificación del texto.

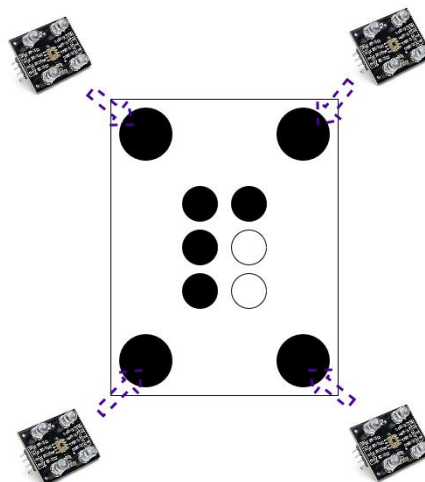
El prototipo será muy sencillo ya que se requiere saber si usted es capaz de reconocer las letras antes de hacer una inversión mas grande para la lectura de páginas completas, por lo que se presentará por medio de tarjetas cada una de las letras.

## POSICIONAMIENTO

En cada una de las tarjetas se tendrán cuatro puntos en cada una de las esquinas para verificar el posicionamiento correcto de la tarjeta, como se muestra a continuación:



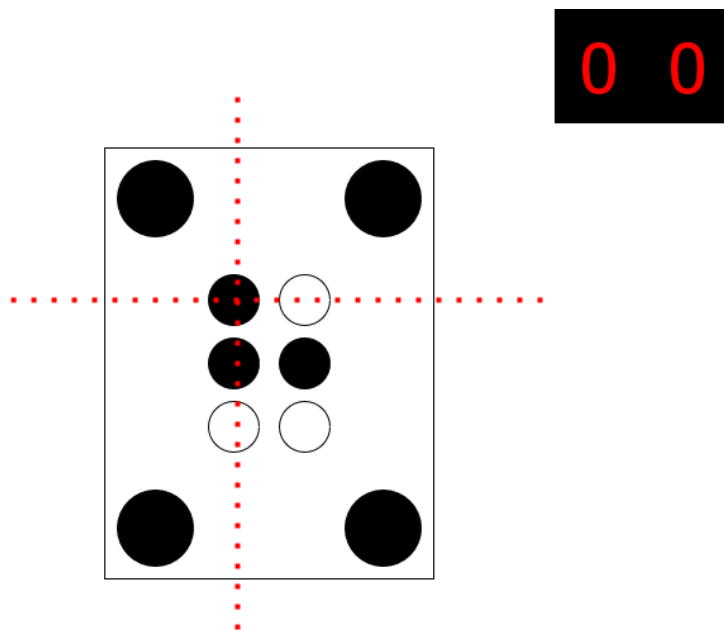
La verificación de posicionamiento se realizará utilizando 4 sensores de color.



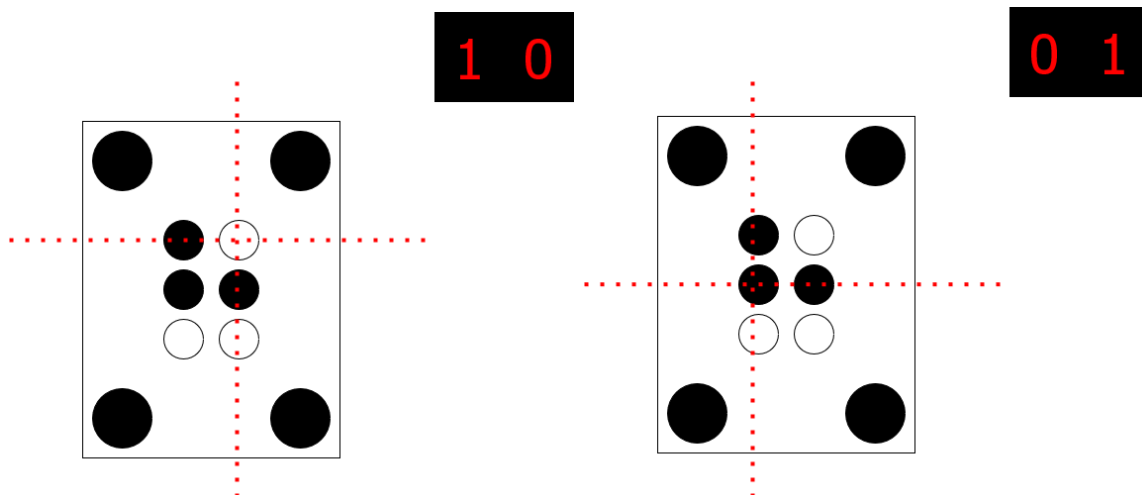
## EJES

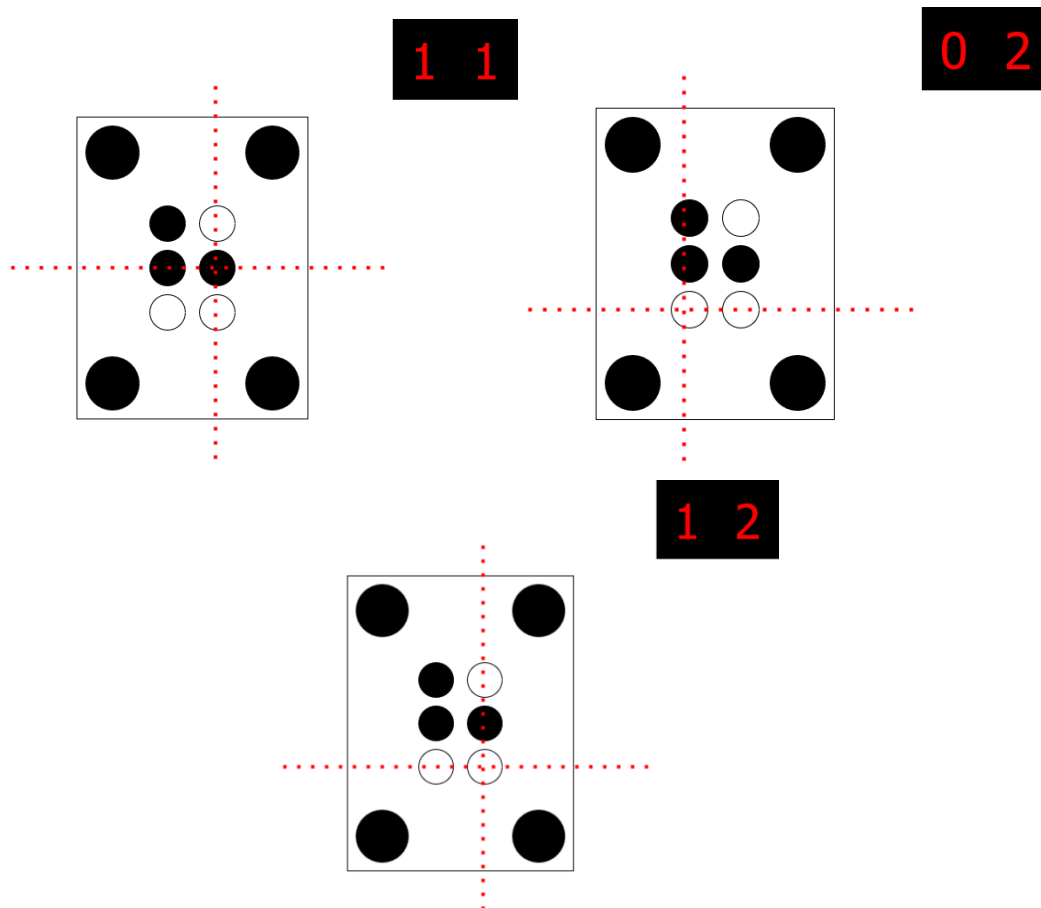
Se deberán tener dos ejes X y Y que se encargarán de mover otro sensor de color que será el encargado de verificar si existen los puntos en el espacio determinado para cada letra, estos ejes deberán ser creados utilizando motores Stepper.

Además, se requiere la implementación de un contador creado utilizando lógica secuencial, siendo los pulsos de reloj generados por el Arduino cada vez que se mueve el eje a una nueva posición.



A continuación, se muestra como debe ser el desplazamiento:





## MEMORIA RAM

Se deberá crear una memoria RAM en que se utilizará como “heap” para almacenar de manera temporal hasta 8 letras, previamente a ser recibidas por la computadora.

**HEAP:** Es una estructura dinámica de datos utilizada para almacenar datos en ejecución. A diferencia de la pila de ejecución que solamente almacena las variables declaradas en los bloques previo a su ejecución, el heap permite reservar memoria dinámicamente, es decir, es el encargado de que la «magia» de la memoria dinámica ocurra. Las variables globales y estáticas también son almacenadas en él.

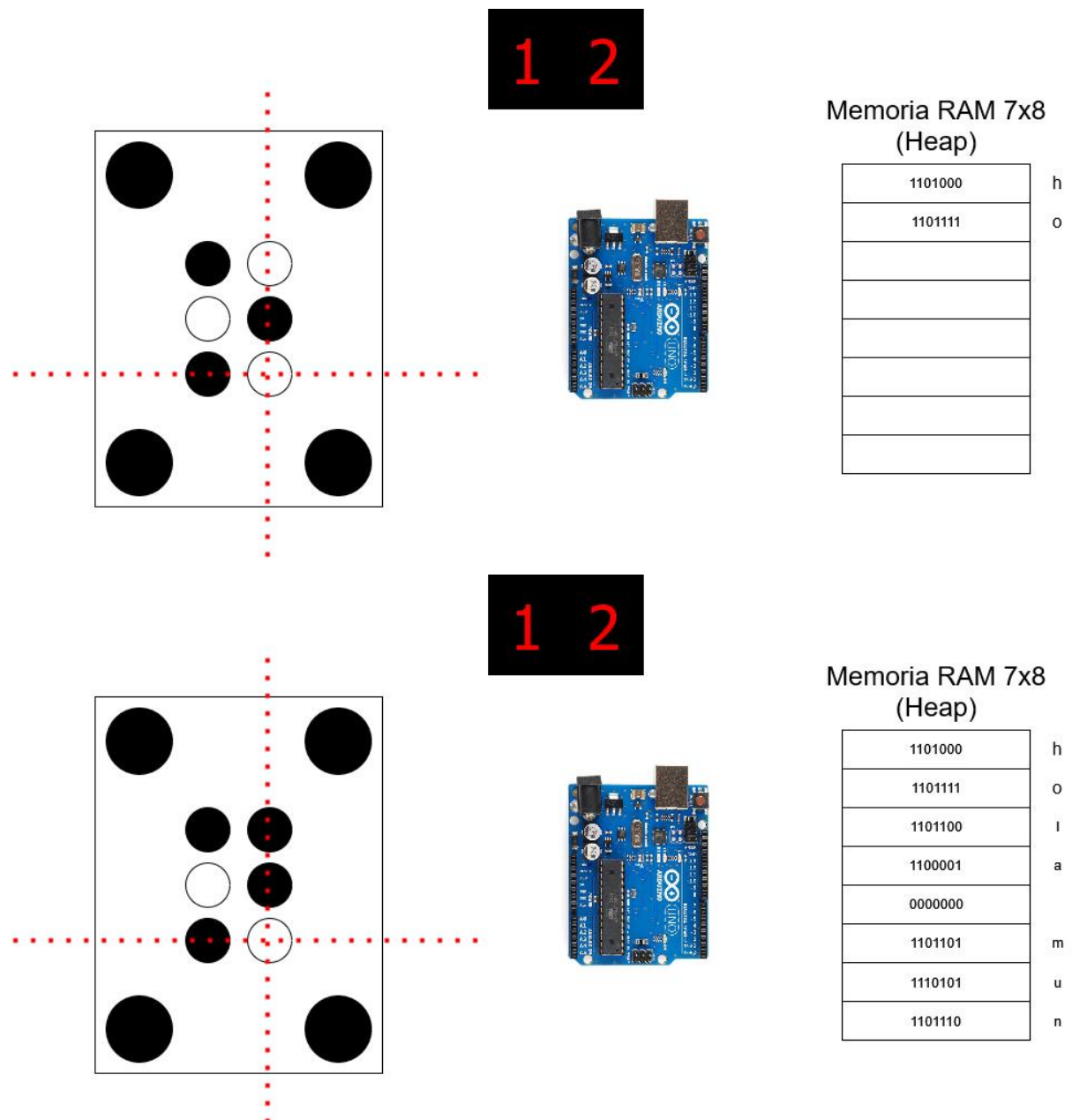
La memoria RAM deberá ser creada dentro de Proteus, por lo que utilizando comunicación serial se deberá enviar el dato desde el Arduino físico a otro Arduino dentro de Proteus que almacenará el dato en la memoria.

El heap almacenará en binario el ASCII equivalente a la letra que ha sido leída por los sensores.

a = 97 en ASCII = 110 0001

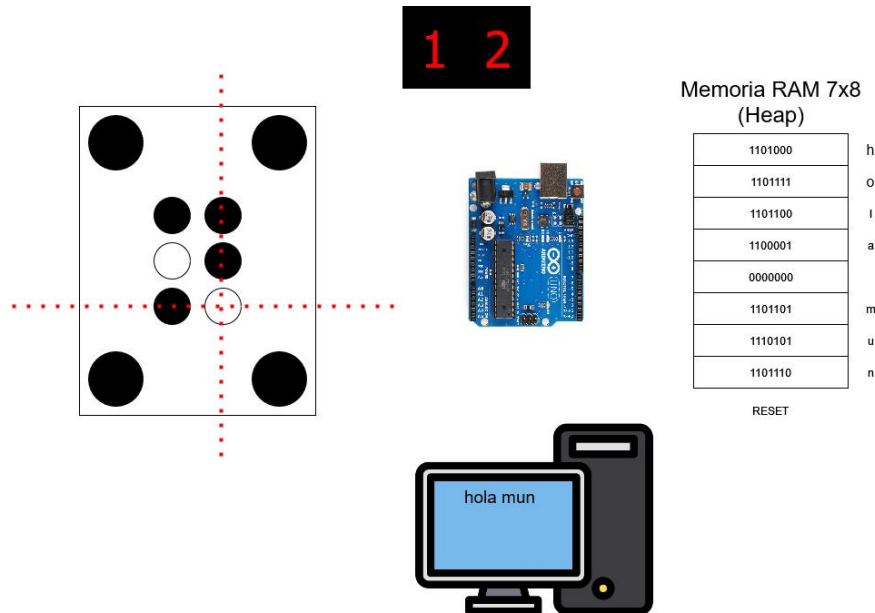
...

z = 122 en ASCII = 111 1010

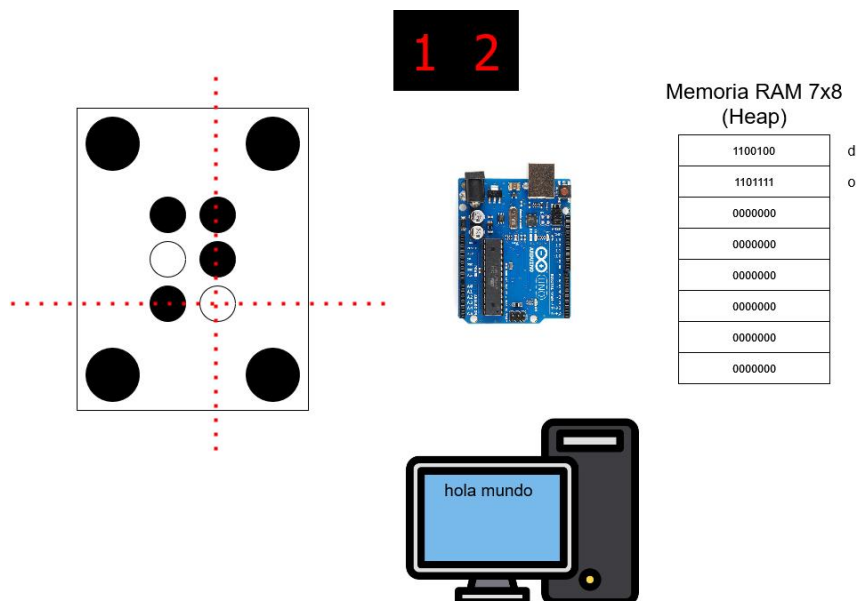


## PROGRAMA DE CONTROL

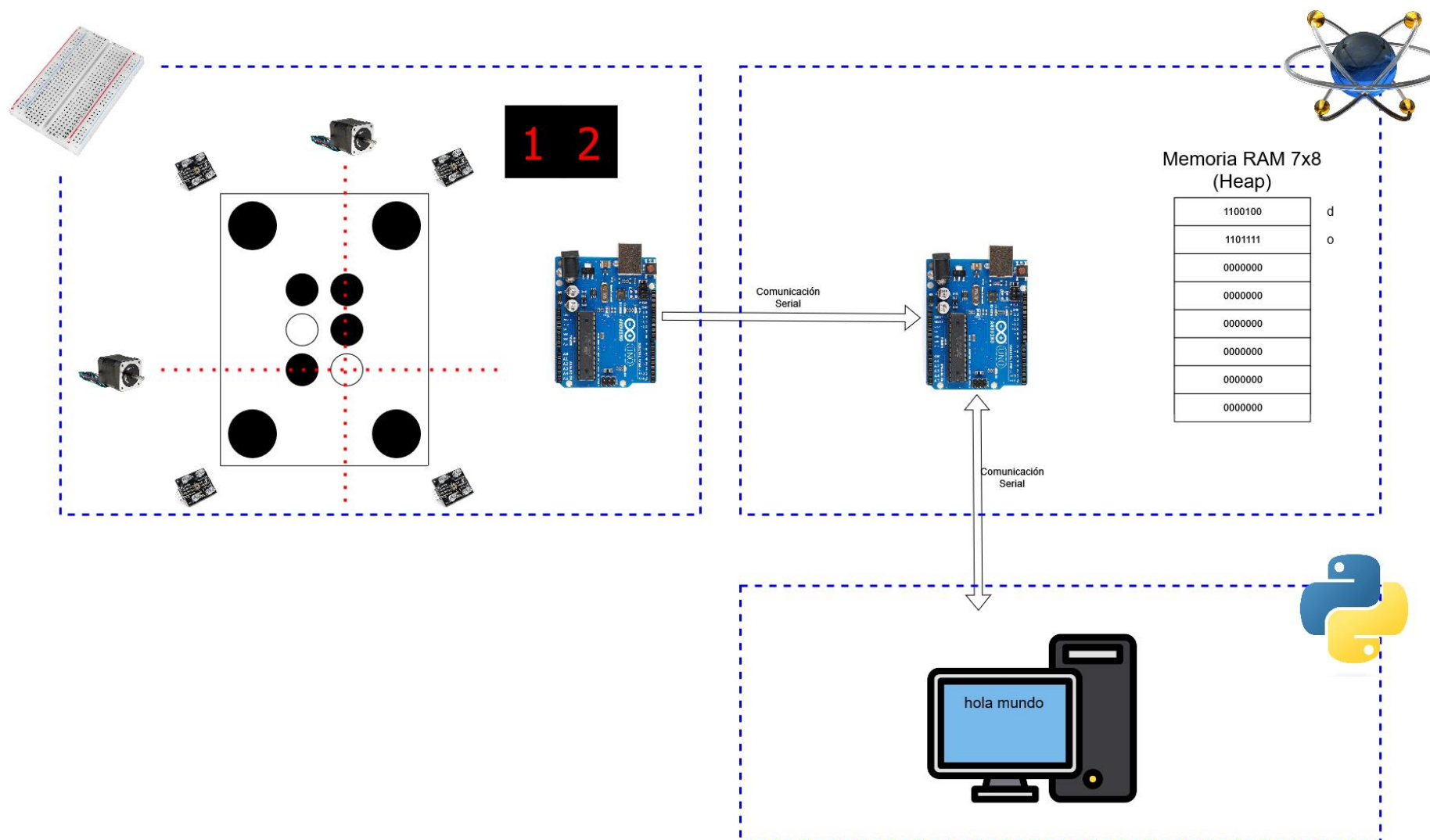
Deberá crear un programa en el lenguaje de programación de su preferencia que será el encargado de leer la información que será enviada desde el Arduino en Proteus al llenarse el heap, luego al asegurarse que los datos fueron recibidos de forma correcta desde el programa se le deberá indicar al Arduino que borre todos los datos del heap y el puntero vuelva a la primera posición de memoria para continuar con la lectura Al llenarse el heap, el Arduino enviará a



Al seguir realizando lecturas en dado caso ya no se requieran leer más tarjetas por medio de un botón se deberá suspender la lectura y enviar las letras recopiladas.



1 2





## **METODOLOGÍA DE TRABAJO**

La práctica debe realizarse en los grupos asignados en las prácticas anteriores.

- Se debe de simular toda la práctica en un solo archivo de Proteus.
- La documentación por entregar deberá ser en forma digital y deberá contener de manera obligatoria el siguiente orden y estar bien identificada cada una de las secciones:

1. Carátula
2. Índice
3. Introducción
4. Descripción del Problema
5. Lógica del Sistema
6. Funciones Booleanas y Mapas de Karnaugh
7. Diagramas de Estado
8. Diagramas del Diseño del Circuito
9. Código comentado
10. Equipo Utilizado
11. Diagramas con Explicación
12. Presupuesto
13. Conclusiones
14. Recomendaciones

### **NOTA:**

Para tener derecho a calificación de documentación, deberán entregar simulación completa del circuito en Proteus.

## CONSIDERACIONES

- El proyecto será calificado sobre el 100% y se estará preguntando a los integrantes sobre el desarrollo de este, de no responder correctamente se restará un cierto porcentaje a la nota obtenida del integrante cuestionado, para asegurar que todos hayan participado en dicha práctica.
- Todos los integrantes deben estar presentes durante la calificación de lo contrario se asumirá que no trabajaron y tendrán nota 0.
- Se penalizará con un porcentaje de la nota obtenida, por no intentar realizar un circuito en físico.
- Se penalizará con un porcentaje de la nota obtenida si la simulación en Proteus se encuentra en distintos archivos.
- Se revisará detalladamente cada circuito si se llegaron a encontrar integrados no permitidos no se podrá seguir con la calificación y tendrán nota 0, todos los integrantes sin excusa alguna.
- Queda prohibido el uso de Logic State.
- Si se detectan copias tendrán nota de 0 y serán reportadas a la Escuela de sistemas.
- Usuario de GitHub al cual se debe de añadir como colaborador al auxiliar:
  - PikaGuty

## ENTREGABLES

Link de drive con los siguientes documentos:

- Un archivo con extensión. pdsprj que contenga el circuito combinacional.
- Documentación.
- Se requiere entregar el enlace al repositorio grupal de GitHub con la siguiente nomenclatura: PF\_ORGA\_G# el cual contendrá todo lo mencionado anteriormente.

## FORMA DE ENTREGA

Mediante UEDI, subiendo el enlace del repositorio. Solamente una persona del grupo debe realizar la entrega.

**Nota:** La fecha límite de entrega es el 27 de junio de 2024, antes de las 23:59.