

VI. LISTA DE COMPONENTES

- Simulador BrainBox
- Proto
- Resistencias
- Cables
- Pulsadores
- Raspberry pi modelo B

VII. MAPA DE VARIABLES

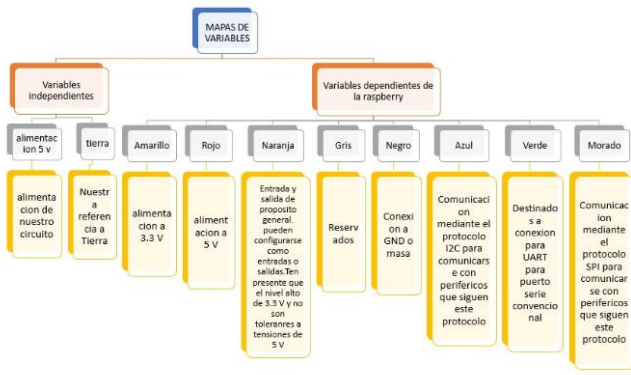


Fig. 6. Variables globales

VIII. EXPLICACIÓN DEL CÓDIGO FUENTE

Uno de los accesorios estrella de la Raspberry Pi es su cámara. Con ella se pueden realizar numerosos proyectos, en este informe se presenta el ejemplo de cómo utilizar este módulo de cámara y que acciones nos permite realizar la raspberry pi. Para este diseño solo presentaremos la simulación, pero es conveniente que si se requiere realizar el proyecto en físico se necesita un módulo de cámara para la raspberry, la cual debe ser una cámara IP para poder monitorear remotamente mediante la raspberry.



Fig. 8. Placa Raspberry

En la imagen se observa un modelo de raspberry en donde encerrado en recuadro rojo observamos la entrada del módulo del cámara conocido como puerto J3.



Fig. 9. Placa Raspberry

Aquí podemos ver ya un módulo de cámara conectado al puerto J3.

Una vez explicado la ubicación donde se conectaría la cámara nos centraremos en la simulación. Para ello utilizamos el simulador BrainBox el cual nos presenta ya un modelo programado de cámara para la raspberry, al modelo de la cámara podemos añadirle una dirección IP de cualquier cámara que tengamos acceso y siempre y cuando sea en formato mjpeg. Nosotros utilizaremos la web insecam.com para obtener una IP de una cámara pública, la cual representará nuestra entrada de datos.



Fig. 10. Ejemplo

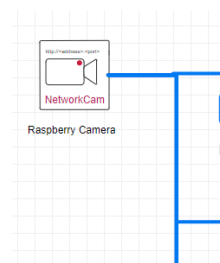


Fig. 11. Ejemplo demostrativo

Este es el modelo programado de la cámara para nuestra raspberry

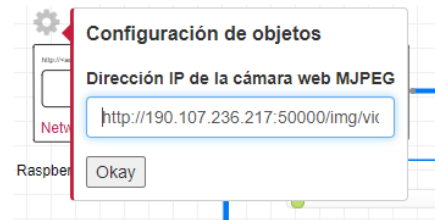


Fig. 12. Ejemplo ingreso de ip

En donde insertamos la dirección IP de nuestra cámara pública.

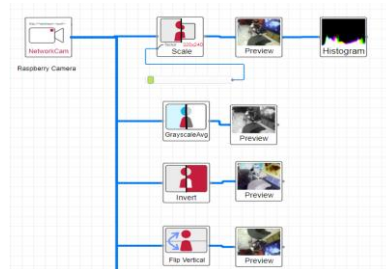


Fig. 13. Ejemplo de ordenadores conectados

En esta parte podemos ver diferentes bloques ya programados que representan el poder que tiene la raspberry para el proceso de imágenes. Explicaremos cada una de ellas:

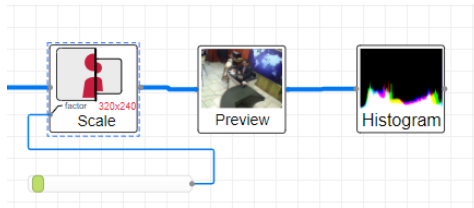


Fig. 14. Ejemplo

Al inicio cuando entran los datos de las imágenes proporcionadas por nuestra cámara IP, es necesario escalar a una resolución adecuada, dicha resolución dependerá del dispositivo en el que queramos reproducir las imágenes y la calidad de las imágenes que nos da la cámara de la raspberry. También podemos realizar un Histograma de imagen que, en un contexto de procesamiento de imágenes, el histograma de una imagen normalmente se refiere a un histograma de los valores de intensidad de píxeles. Este histograma es un gráfico que muestra el número de píxeles de una imagen en cada valor de intensidad diferente que se encuentra en esa imagen. Para una imagen en escala de grises de 8 bits, hay 256 intensidades posibles diferentes, por lo que el histograma mostrará gráficamente 256 números que muestran la distribución de píxeles entre esos valores de escala de grises.

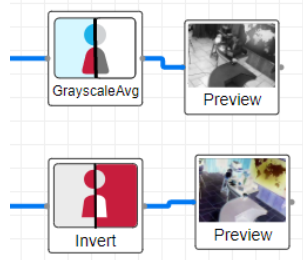


Fig. 15. Ejemplo

Convertidor de escala de grises promedio: El método promedio es el método más simple de color a escala de grises. Solo debes tomar la media de tres colores. Dado que es una imagen RGB, significa que debe agregar r con g con b y luego dividirlo por 3 para obtener la imagen en escala de grises deseada.

Inversión de color: La inversión de color, también conocida como efecto negativo, es uno de los efectos más fáciles de lograr en el procesamiento de imágenes. La inversión de color se logra restando cada valor de color RGB del valor máximo posible (generalmente 255). La inversión puede ser necesaria para realizar algunas operaciones, como operaciones morfológicas. Por ejemplo, la erosión reduce los límites de las regiones de blanco / primer plano, por lo que importa qué píxeles son blancos / primer plano.

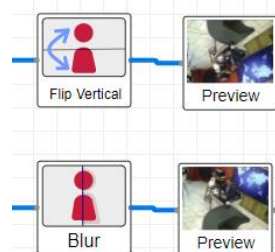


Fig. 16. Ejemplo

Voltear filtro vertical: Un giro (efecto espejo) se realiza invirtiendo los píxeles horizontal o verticalmente.

Filtro de desenfoque gaussiano: Ejecuta un desenfoque Gaussian con el parámetro de nivel que especifica la extensión del desenfoque. Es un efecto ampliamente utilizado en software de gráficos, generalmente para reducir el ruido de la imagen y reducir los detalles.

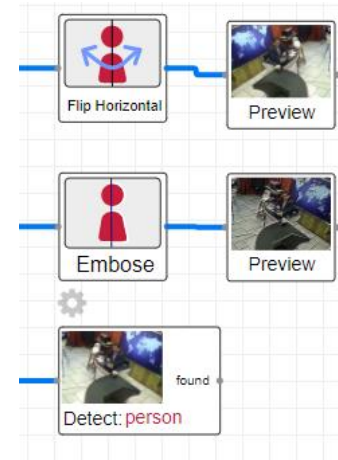


Fig. 17. Ejemplo

Flip horizontal: Un giro (efecto espejo) se realiza invirtiendo los píxeles horizontal o verticalmente.

Filtro de relieve: La función EMOSS aplica un operador de convolución a una matriz de imágenes 2D para generar una matriz que contiene valores de diferencia que representan bordes en la imagen original. Este proceso imparte una apariencia en relieve a la imagen.

Detector de objetos: Modelo de detección de objetos que tiene como objetivo localizar e identificar un objeto en una sola imagen.

IX. DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Para poder ingresar al simulador y revisar la simulación de la cámara de la raspberry ingresamos al siguiente link:

<http://www.brainbox-demo.de/circuit/?shared=5eQD@Ogef.brain> se nos abrirá directamente la simulación.

En el caso que nos pidiera usuario y contraseña ingresamos estos datos:

Usuario: Digitales

Clave: h5xCbP9V4

Y nos dirigimos al apartado de proyectos guardados:

Y elegimos el archivo de nombre raspberry

Aquí ya tenemos la simulación completa, la podemos simular y observar los diferentes filtros que nos ofrece el simulador, los cuales podemos programar en la raspberry y utilizar las diferentes librerías que nos ayudan a usar dichos filtros.

Para poder variar los datos y utilizar otra cámara IP nos dirigimos a la siguiente página:

<http://insecam.org/en/bycountry/EC/>



Fig. 18. Página insecam

En esta página podremos escoger otra dirección IP de cualquier cámara publica, siempre y cuando se transmita en formato mpjpeg.



Fig. 19. enlace para dirección ip

Debemos presionar en configuración y podremos pegar el link con una nueva dirección IP.

X. APORTACIONES

En este apartado realizaremos un ejemplo un poco más interactivo en el que revisaremos las dos filas de pines que lleva consigo la raspberry y las usaremos para poder controlar el encendido y apagado de leds.

Para ello utilizamos una raspberry modelo B.

Ahora observamos la distribucion de pines de la raspberry:

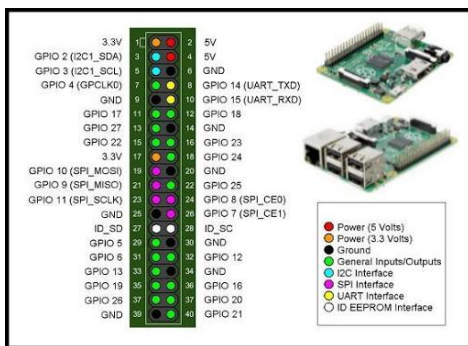


Fig. 20. Pines de la raspberry

Explicacion del codigo:

```
#!/usr/bin/env python    # Necesario para el interprete
Python
"""
Programa simple que enciende 4 leds conectados
a una Raspberry PI en modo secuencial.
Bryan Carrillo
```

Uso:

1. Accionar el pulsador 1 para activar la secuencia
2. Accionar el pulsador 2 para terminar el programa

```
import RPi.GPIO as io      # Import libreria de I/O (entradas /
                             salidas)
import time                # Import libreria para los retardos

led1 = 9                   # led 1 conectado al gpio 9 del micro
led2 = 8                   # led 2 conectado al gpio 8 del micro
led3 = 7                   # led 3 conectado al gpio 7 del micro
led4 = 4                   # led 4 conectado al gpio 4 del micro

button1 = 11               # boton 1 conectado al gpio 11 del
                             micro
button2 = 10               # boton 2 conectado al gpio 10 del
                             micro

io.setmode(io.BCM)         # modo in/out pin del micro
io.setwarnings(False)      # no señala advertencias de pin ya
                             usados

io.setup(led1,io.OUT)       # set pin led 1 como salida
io.setup(led2,io.OUT)       # set pin led 2 como salida
io.setup(led3,io.OUT)       # set pin led 3 como salida
io.setup(led4,io.OUT)       # set pin led 4 como salida

io.setup(button1,io.IN)     # set pin pulsador 1 como entrada
io.setup(button2,io.IN)    # set pin pulsador 2 como entrada

for a in range(100):       # Ciclo de espera hasta que se
                             acciona
    time.sleep(0.5)         # el pulsador 1
    if not io.input(button1):
        break               # Con el pulsador 1 accionado (a masa)
                             # termina el ciclo de espera

while io.input(button2):    # Ciclo hasta que se acciona el
                             pulsador 2
    io.output(led4,0)        # Apaga el led 4
    io.output(led1,1)        # Enciende el led 1
    time.sleep(0.1)          # Espera medio segundo

    io.output(led1,0)        # Apaga el led 1
    io.output(led2,1)        # Enciende el led 2
    time.sleep(0.1)          # Espera otro medio segundo

    io.output(led2,0)        # Apaga el led 2
    io.output(led3,1)        # Enciende el led 3
    time.sleep(0.1)          # Espera otro medio segundo

    io.output(led3,0)        # Apaga el led 3
    io.output(led4,1)        # Enciende el led 4
    time.sleep(0.1)          # Espera otro medio segundo

    io.output(led4,0)        # Apaga el led 4 antes de terminar el
                             programa
```


Abrimos el terminal de raspbian y ejecutamos nuestro programa:

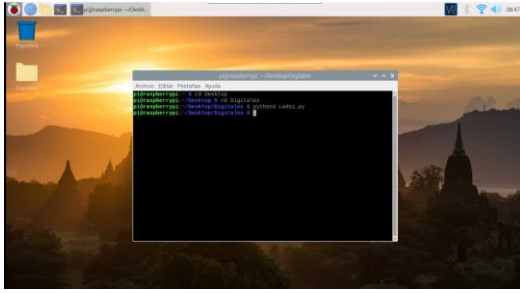


Fig. 21. cmd

Y observamos como comienzan a encenderse los leds:

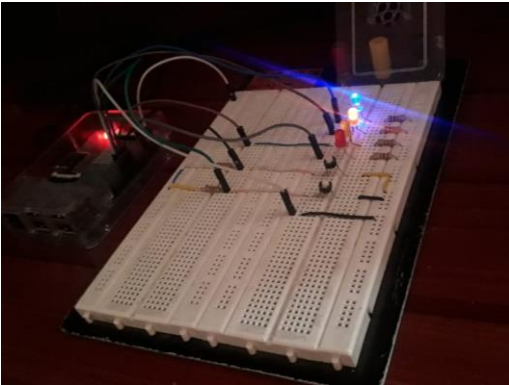


Fig. 20. Protoboard

XI. CONCLUSIONES

Entender cómo se distribuyen los diferentes pines o puertos tanto de salida y para qué sirve cada uno resulta muy sencillo. Existen diversos artículos en el internet que lo explican detalladamente y de una manera interactiva, lo cual facilita el proceso de entendimiento para principiantes.

El simulador BrainBox trabaja con dispositivos programados para poder simular circuitos digitales, en el apartado de la raspberry no cuenta con gran variedad en el caso de periféricos, pero los que se encuentran ya en uso trabajan mediante bloques, lo cual resulta interactivo y facilita el aprendizaje a principiantes.

Para nuestra aportación en físico observamos que las técnicas de programación en Python pueden ser muy simples o muy sofisticadas (por ejemplo, con la programación orientada a objetos). En este proyecto se trató de reducir la complejidad para ser lo más didácticos posibles. Python, Raspberry y Linux son argumentos muy extensos sobre los cuales pueden encontrar toneladas de artículos en Internet y también tantos libros.

XII. RECOMENDACIONES

Es importante conocer la estructura de hardware de la raspberry pi, para evitar daños en la realización de proyectos.

Se recomienda tener conocimientos en programación de lenguaje Python para poder empezar a trabajar en la raspberry.

Es recomendable al instalar un sistema operativo en la raspberry, contar con un teclado y mouse con conexión usb o inalámbricos.

Es preciso planificar un cronograma con diagramas de Grant en las diferentes aplicaciones que existen y para el desarrollo se recomienda el software Project.

XIII. ANEXOS

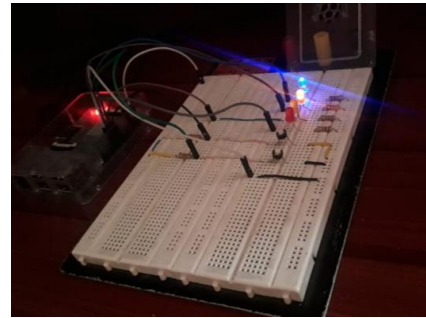


Fig. 21. Protoboard

XIV. HOJA TÉCNICA

Ficha Técnica de la raspberry

Physical Specifications

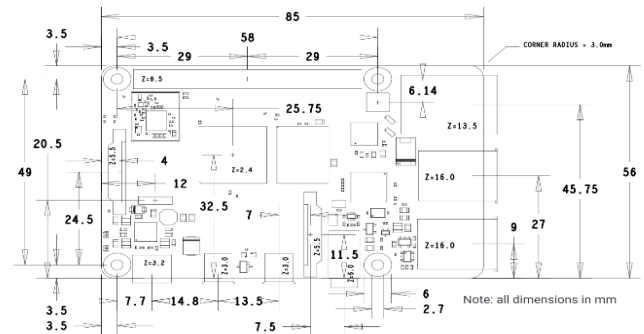


Fig. 22. Anexo1

REFERENCIAS

- [1] R. Siliceo. (2018). *Algoritmo de las operaciones aritmeticas aplicadas a los codigos binarios, octal y hexadecimal con sus respectivas conversions* Ciudad de Mexico
- [2] T. Floyd, "Fundamentos de sistemas digitales", 6ta ed. Ed. Madrid: Pearson, 2006. 15–64.
- [3] D. Alulema, (2020) "*Circuitos digitales*" Quito, Ecuador
- [4] A. Ricoy. (2020,06). *App inventor en español* Available: <https://sites.google.com/site/appinventormegusta/primeros-pasos>