

## Análisis y Diseño:

Tabla:

Archivos Fuente	Nivel	Subrutinas		Porcentaje de Código	
Comunicación _Ar_RB.ino	Alto	Entradas	Datos de temperatura, humedad y luminosidad.	35%	
		Función de subrutina	<b>Setup:</b> coloca todos los pines del 3-10 como salida del bus, y se comienza la comunicación serial. <b>Loop:</b> Lee sensor de luz (analogRead), lee sensor de temperatura (utiliza la librería dht.read11) y por ultimo lee el sensor de humedad (con la misma librería). displayBinary: método para desplegar binarios en un bus de 8 bits. diplayBinary2: método para desplegar de 0-3 el dato que se esta enviando.		
		Salidas	Leds de los pines 2-11. Muestra dato de luminosidad. Muestra dato de temperatura. Muestra dato de humedad.		
Phys_to_virt.c	Alto	Permite mapear una dirección física y guardar el espacio en memoria.		5%	
Pixel.c		Dibuja pixeles en coordenadas x,y			
BienvePrin.s	Bajo	Matriz de imagen en 8 bits de la planta más pequeña.		60%	
VENTANA1.s		Matriz de la imagen en 8 bits de la planta más grande.			
VENTANA2.s		Matriz de la imagen en 8 bits de la planta normal.			
VENTANA3.s		Matriz de la imagen en 8 bits de fondo			
Dibujar.s		Entradas	Dirección de memoria, coordenadas		
		Función de subrutina	<b>Dibujar:</b> Recibe las posiciones temporales de las imagen, al igual que las coordenadas iniciales en x,y de las mismas. <b>Recorrer:</b> manda a llamar a la dirección que le otorga el archivo pixel.c ara dibujar <b>Loop3:</b> pinta en x,y hasta que las coordenadas temporales sean iguales a las x,y originales. También el contador se reinicia. <b>Fin1:</b> termina el proceso de pintar		
	Salidas	Pixeles pintados			

getGpio.s		Obtiene la dirección de memoria		60%
Gpio0.s		Contiene las rutinas que permiten la manipulación de pines GPIO en la Raspberry.		
Main		Ejecutable del proyecto		
		Entradas	Puertos de GPIO: 21,15,18,23,24,25,08,07,	
		Función de subrutina	<p><b>Main:</b> manda a llamar a la subrutina <b>getScreenAddr</b> que tiene la dirección de memoria. Y llama a la subrutina <b>GEtGpioAddress</b>, que permite el uso de los pines GPIO:</p> <p><b>SetGpioFunction:</b> activa el pino deseado.</p> <p><b>terminarPrograma:</b> enciende led que muestra que el programa esta encendido, de lo contrario termina todo proceso.</p> <p><b>inicioPrograma:</b> llama a DibujarVENTANA 3 que es el inicio, e incluye función para revisar el estado de los botones.</p> <p><b>Botsalir:</b> imprime mensaje de despedida</p> <p><b>noSalir:</b> se encarga de leer los datos</p> <ul style="list-style-type: none"><li>• <b>bit00:</b> revisa si el bit ingresado es 00, de ser así llama a <b>sibit00</b></li><li>• <b>siBit00:</b> revisa que para luz e sea distinto de 00 y para temperatura que sea 10</li><li>• <b>noBit00:</b> revisa que si es 01 entonces llama a humedad, de lo contrario se regresa a la imagen principal.</li></ul> <p><b>Temperatura:</b> manda a llamar a la subrutina para leer el dato.</p> <ul style="list-style-type: none"><li>• <b>Bit01_t:</b> se revisa el primer bit ingresado. <b>siEsUnot:</b> muestra mensaje que si es uno, de lo contrario <b>noEsUnot</b> se ejecuta.</li><li>• <b>Bit02_t:</b> se revisa el segundo bit, si el resultado es 2, entonces se ejecuta <b>siEsDost</b> e imprime mensaje de lo contrario se muestra <b>noEsDost</b></li><li>• <b>Bit03_t:</b> revisa el tercer bit, es decir que el valor sea 4.</li><li>• <b>Bit04_t:</b> revisa el cuarto bit, es decir que el valor sea 8.</li></ul>	
Main.s	Bajo			

Main.s	Bajo	Función de subrutina	<ul style="list-style-type: none"> <li>• <b>Bit05_t</b>: revisa el quinto bit, es decir que el valor sea 16.</li> <li>• <b>Bit06_t</b>: revisa el sexto bit, es decir que el valor sea 32.</li> <li>• <b>Bit07_t</b>: revisa el séptimo bit, es decir que el valor sea 64.</li> <li>• <b>Bit08_t</b>: revisa el octavo bit, es decir que el valor sea 128.</li> </ul> <p><b>compTemp</b>: lee el valor de temperatura, si la esta es &lt; a 25 entonces ejecuta encenderTemperatura que enciende un led, e imprime la imagen de la planta grande. De lo contrario si la humedad es &gt; a 25, se paga el led y se ejecuta NoencenderTemperatura que imprime la planta pequeña.</p> <p><b>Humedad</b>: manda a llamar a la subrutina para leer el dato.</p> <ul style="list-style-type: none"> <li>• <b>Bit01_h</b>: revisa el primer bit, si el valor es 1, entonces se ejecuta <b>siEsUnoh</b> que imprime mensaje de afirmación, de lo contrario se ejecuta <b>noEsUnoh</b>:</li> <li>• <b>Bit02_h</b>: se revisa el segundo bit, si el resultado es 2, entonces se ejecuta <b>siEsDosh</b> e imprime mensaje de lo contrario se muestra <b>noEsDosh</b></li> <li>• <b>Bit03_h</b>: revisa el tercer bit, es decir que el valor sea 4.</li> <li>• <b>Bit04_h</b>: revisa el cuarto bit, es decir que el valor sea 8.</li> <li>• <b>Bit05_h</b>: revisa el quinto bit, es decir que el valor sea 16.</li> <li>• <b>Bit06_h</b>: revisa el sexto bit, es decir que el valor sea 32.</li> <li>• <b>Bit07_h</b>: revisa el séptimo bit, es decir que el valor sea 64.</li> <li>• <b>Bit08_h</b>: revisa el octavo bit, es decir que el valor sea 128.</li> </ul> <p><b>compHum</b>: lee el valor de humedad, si la humedad &lt; a 30 entonces ejecuta encenderHumedad que enciende un led, e imprime la imagen de la planta pequeña. De lo contrario si la humedad</p>	60%
--------	------	----------------------	--	-----

Main.s	Bajo	Función de subrutina	<p>es &gt; a 30, apaga el led y se ejecuta NoencederHumedad que imprime la planta normal.</p> <p><b>Luminosidad:</b> manda a llamar a la subrutina para leer el dato.</p> <ul style="list-style-type: none"> <li>• <b>Bit01_1:</b> revisa el primer bit, si el valor es 1, entonces se ejecuta <b>siEsUno</b> que imprime mensaje de afirmación, de lo contrario se ejecuta <b>noEsUno</b></li> <li>• <b>Bit02_1:</b> se revisa el segundo bit, si el resultado es 2, entonces se ejecuta <b>siEsDos</b> e imprime mensaje de lo contrario se muestra <b>noEsDos</b></li> <li>• <b>Bit03_1:</b> revisa el tercer bit, es decir que el valor sea 4.</li> <li>• <b>Bit04_1:</b> revisa el cuarto bit, es decir que el valor sea 8.</li> <li>• <b>Bit05_1:</b> revisa el quinto bit, es decir que el valor sea 16.</li> <li>• <b>Bit06_1:</b> revisa el sexto bit, es decir que el valor sea 32.</li> <li>• <b>Bit07_1:</b> revisa el séptimo bit, es decir que el valor sea 64.</li> <li>• <b>Bit08_1:</b> revisa el octavo bit, es decir que el valor sea 128.</li> </ul> <p><b>compLum:</b> lee el valor de luz, si esta es &lt; 200 entonces ejecuta encenderLuminosidad que enciende un led, e imprime la imagen de la planta grande. De lo contrario si la humedad es &gt; a 200, entonces apaga el led y se ejecuta NoencederHumedad que imprime la planta normal.</p> <p>Fin: apaga los led conectados a los pines 17,26,13,19</p> <p>-----SUBROUTINAS PARA DIBUJAR-----</p> <p>dibujarBienvePrin: subrutina a la que se llama para dibujar la planta más pequeña.</p>	60%
--------	------	----------------------	---	-----

Main.s	Bajo	Función de subrutina	<ul style="list-style-type: none"> <li>drawBienvePrin: compara x con el ancho de la imagen. Se obtiene la dirección de la matriz y lee el dato de la matriz.</li> <li>filaBienvePrin: revisa que se haya dibujad toda la imagen.</li> </ul> <p><b>dibujarVENTANA1:</b> subrutina a la que se llama para dibujar la planta más grande.</p> <ul style="list-style-type: none"> <li>drawVENTANA1: compara x con el ancho de la imagen. Se obtiene la dirección de la matriz y lee el dato de la matriz.</li> <li>filaVENTANA1: revisa que se haya dibujado toda la imagen.</li> </ul> <p><b>dibujarVENTANA2:</b> subrutina a la que se llama para dibujar la planta más grande.</p> <ul style="list-style-type: none"> <li>drawVENTANA2: compara x con el ancho de la imagen. Se obtiene la dirección de la matriz y lee el dato de la matriz.</li> <li>filaVENTANA2: revisa que se haya dibujado toda la imagen.</li> </ul> <p><b>dibujarVENTANA3:</b> subrutina a la que se llama para dibujar la planta más grande.</p> <ul style="list-style-type: none"> <li>drawVENTANA3: compara x con el ancho de la imagen. Se obtiene la dirección de la matriz y lee el dato de la matriz.</li> <li>filaVENTANA3: revisa que se haya dibujado toda la imagen.</li> </ul>	60%
		Salidas	Puertos de escritura: 4,17,26,19,13,22,27 4: botón de salida, muestra mensaje 17: led de encendido 22 y 27: muestran el dato que se está leyendo (01 para humedad, 10 para temperatura y 11 para luz)	

### Algoritmo Narrativo:

Bien en primer lugar, se realiza la lectura de los sensores, los cuales, en base a una programación en alto nivel, reciben ese dato y es enviado a la Raspberry mediante un bus constituido por 8 bits. (García, 2016)

Una vez en la Raspberry, esta lee el dato y detecta que tipo de dato está recibiendo, si es 01, es porque se trata de humedad, si es 10 se trata de temperatura y si es 11 entonces es porque se trata de luz. Esto se ve reflejado mediante el uso de 2 leds. (Perez, 2016)

Al saber que dato se recolecto, entonces se verificar de cuanto es su valor. Para ello se realización varias subrutinas que lo que hacen es comparar el valor del bit que está ingresando. Por ejemplo, si ingreso un 7, entonces quiere decir que el sensor leyó un valor de 128 en decimal. Una vez detectado el valor del bit ingresado, entonces se hacen las respectivas comparaciones para saber qué imagen imprimir de acuerdo al dato recolectado. (Glare, 2016)

Para ello actúa entonces la interfaz gráfica, y es que es la encargada en este caso de llevar a cabo la simulación para saber que pudo llegarle a pasar a la planta en dado caso este expuesta a dichos valores recolectados. Entonces si la planta resulto con una temperatura más alta de 25, la interfaz mostrara la imagen de una planta en buenas condiciones, de lo contrario mostrara una planta en decadencia. Este mismo procedimiento se repite para los valores de humedad y luminosidad.

Por último, los leds que quedasen encendidos, se apagan y la simulación finaliza.

### Bibliografía

- García, A. (20 de Noviembre de 2016). *Panama Hitek*. Obtenido de Panama Hitek:  
<http://panamahitek.com/libreria-dht-para-usar-sensor-de-humedad-y-temperatura-con-arduino/>
- Glare, C. (20 de Noviembre de 2016). *RoboLogs*. Obtenido de RoboLogs:  
<http://robologs.net/2014/04/12/tutorial-de-raspberry-pi-gpio-y-python-i/>
- Perez, M. (20 de Noviembre de 2016). *GeekyTheory*. Obtenido de GeekyTheory:  
<https://geekytheory.com/arduino-raspberry-pi-raspduino>