

## **MICROCONTROLADORES**

### **Práctica No. 13. Medidor de Humedad y Temperatura usando un DHT11.**

#### **1. Objetivo**

- Display LCD alfanumérico 2x16.
- Modulo PCF8574.
- Un sensor DHT11.

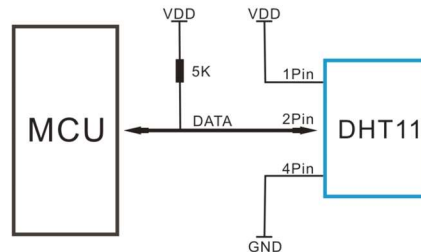
#### **2. Material y Equipo.**

- Computador o laptop con el STM32CubeIDE.
- Un Display LCD alfanumérico.
- Un módulo PC8574.
- Un sensor DHT11.

#### **3. Marco de Referencia.**

El sensor DHT11 es un sensor de humedad y temperatura digital el cual incorpora un sensor resistivo de humedad y para la temperatura un sensor NTC ambos conectados a un microcontrolador de 8 bits el cual se encarga de tomar las mediciones de humedad, temperatura y el protocolo de comunicación. El DHT11 se alimenta con un voltaje de entre 3V a 5V como voltaje recomendado, una vez alimentado el sensor iniciará en modo de bajo consumo de voltaje cuando el sensor detecte una secuencia de inicio saldrá de este estado y pasará al estado activo e iniciará la comunicación con el microcontrolador y transferirá los datos de humedad y temperatura.

### 3.1 Circuito recomendado.

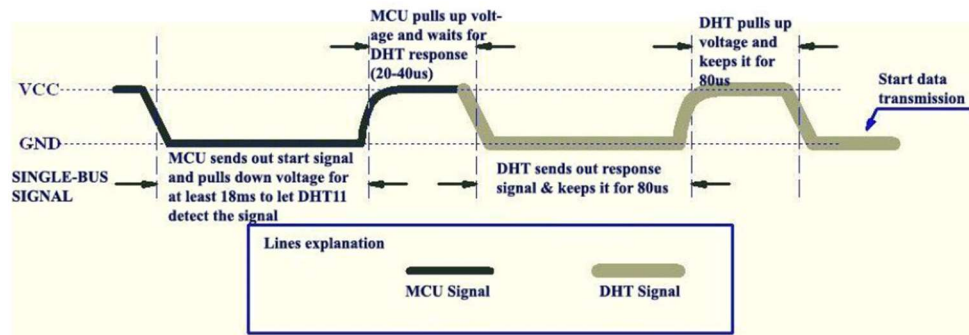


El circuito de arriba es el circuito recomendado para conectar un sensor DHT11 y un microcontrolador en este caso el STM32F103C8T6, si la distancia entre el microcontrolador y el sensor es menor a 20 mts. se recomienda una resistencia de pull-up de 5K en la línea de datos, si la distancia es mayor a 20 mts. se tiene que calcular dicha resistencia.

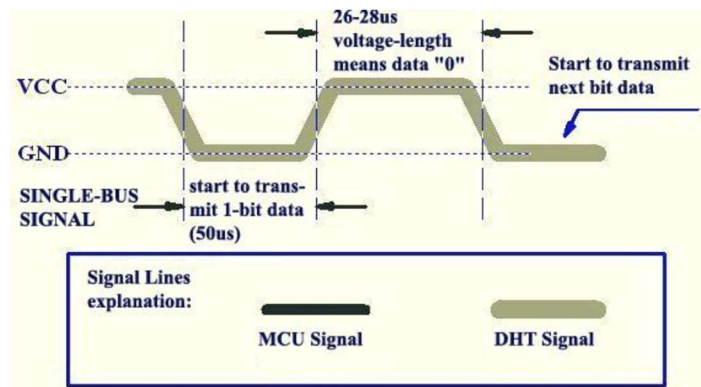
### 3.2 Protocolo de Comunicación.

El protocolo de comunicación es de una sola línea entre el STM32 y el sensor DHT11. Con una sola línea de datos se hace la transferencia de los datos y la sincronización con el sensor. El total de datos transferidos entre el DHT11 y el STM32 es de 40 bits que es la humedad relativa dos bytes (parte entera y parte fraccionaria), temperatura dos bytes (parte entera y parte fraccionaria) y por último un byte del checksum para verificar si el dato recibido es correcto.

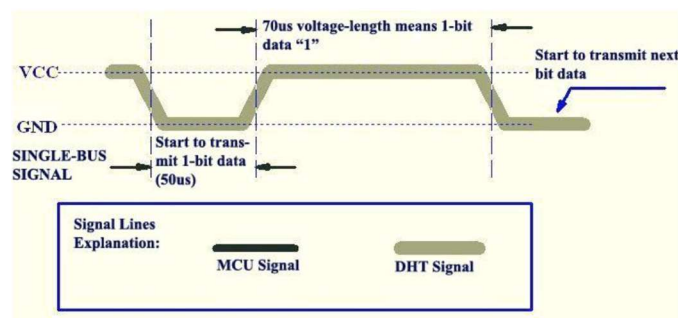
La comunicación entre el STM32 y el DHT11 empieza con una secuencia de inicio y una respuesta del DHT11. La secuencia de inicio consiste de un pulso bajo con una duración de 18 ms y un pulso en alto con una duración de entre 20  $\mu$ s a 40  $\mu$ s. Una vez que el DHT11 detecta la secuencia de inicio el sensor bajara la línea de datos por 80  $\mu$ s, después el sensor enviara un pulso de bajo a alto con una duración de 80  $\mu$ s antes de enviar la información de humedad y temperatura.



Una vez que el DHT11 envía la respuesta de que está listo para transferir la información esta se enviara bit a bit con el siguiente formato, tanto para el uno como el cero lógico se empieza con un pulso en bajo (bit de inicio) con una duración de 50  $\mu$ s, lo siguiente es enviar el ancho de pulso correspondiente al cero o al uno. Para un cero lógico se envía un pulso en alto con una duración de entre 26  $\mu$ s a 28  $\mu$ s.



Para un uno lógico se envía un pulso en alto con una duración de 70  $\mu$ s.

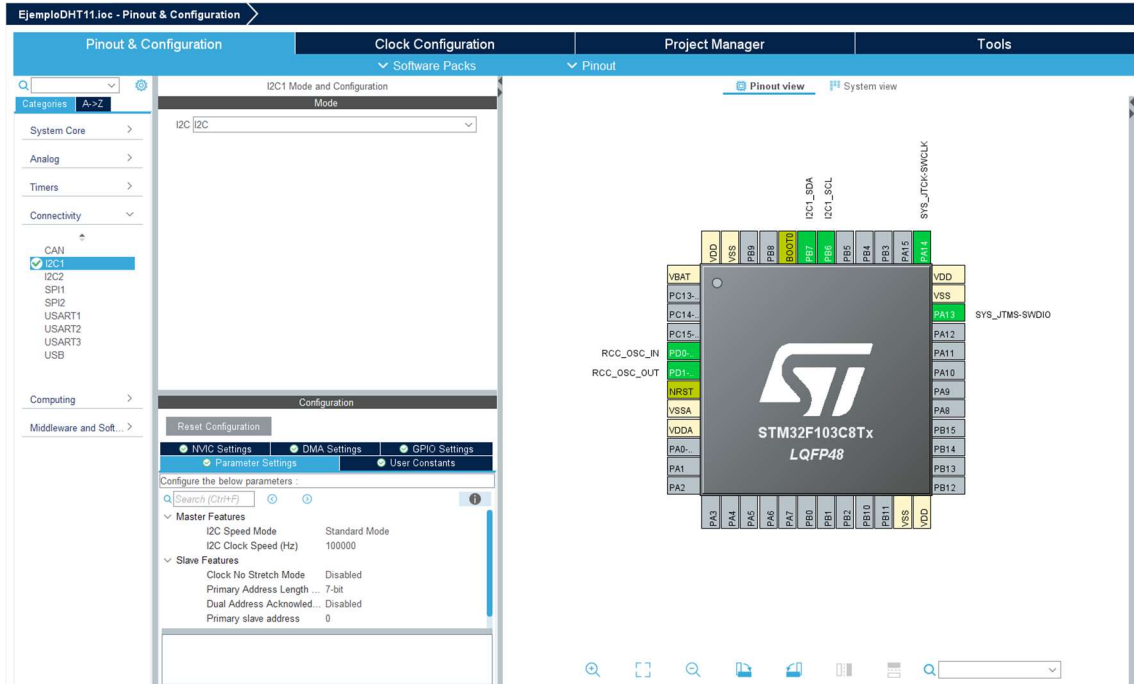


Cuando la señal de respuesta del DHT11 es siempre alto significa que el sensor no está respondiendo correctamente y se tiene que verificar las conexiones entre el STM32 y el DHT11. Cuando el ultimo bit es transmitido la señal de datos se mantendrá en bajo por 50  $\mu$ s y después pasará alto automáticamente debido a la resistencia de pull-up y el DHT11 entrar en modo de bajo consumo y la línea de datos pasará a estado libre.

NOTA: No hay una función HAL para microsegundos por lo que se utilizara el TIM2 para crear una función delay de microsegundos.

#### 4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.





El código de la práctica es el siguiente.

```
1 #include "main.h"
2 #include "i2c.h"
3 #include "tim.h"
4 #include "gpio.h"
5 #include "lcd.h"
6 #include <stdio.h>
7
8 void SystemClock_Config(void);
9
10 void delay (uint16_t time)
11 {
12     __HAL_TIM_SET_COUNTER(&htim2, 0);
13     while ((__HAL_TIM_GET_COUNTER(&htim2)) < time);
14 }
15
16 void Display_Temp (float Temp)
17 {
18     char str[20] = {0};
19     Lcd_Gotoxy(1, 1);
20
21     sprintf(str, "TEMP:- %.2f ", Temp);
22     Lcd_Print(str);
23     Lcd_Print("C");
24 }
25
26 void Display_Rh (float Rh)
27 {
28     char str[20] = {0};
29     Lcd_Gotoxy(1, 2);
30
31     sprintf(str, "RH:- %.2f ", Rh);
32     Lcd_Print(str);
33     Lcd_Print("%");
34 }
35
36 uint8_t Rh_byte1, Rh_byte2, Temp_byte1, Temp_byte2;
37 uint16_t SUM, RH, TEMP;
38
39 float Temperature = 0;
40 float Humidity = 0;
41 uint8_t Presence = 0;
42
43 void Set_Pin_Output (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin)
44 {
45     GPIO_InitTypeDef GPIO_InitStructure = {0};
46     GPIO_InitStructure.Pin = GPIO_Pin;
47     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
48     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
49     HAL_GPIO_Init(GPIOx, &GPIO_InitStructure);
50 }
51
52 void Set_Pin_Input (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin)
53 {
54     GPIO_InitTypeDef GPIO_InitStructure = {0};
55     GPIO_InitStructure.Pin = GPIO_Pin;
56     GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
57     GPIO_InitStructure.Pull = GPIO_PULLUP;
58     HAL_GPIO_Init(GPIOx, &GPIO_InitStructure);
59 }
60
61 #define DHT11_PORT  GPIOA
62 #define DHT11_PIN  GPIO_PIN_1
```





```
64 void DHT11_Start (void)
65 {
66     Set_Pin_Output (DHT11_PORT, DHT11_PIN); // poner el pin como salida
67     HAL_GPIO_WritePin (DHT11_PORT, DHT11_PIN, 0); // iniciar el pin en 0
68     HAL_Delay(18); // esperar 18ms
69     HAL_GPIO_WritePin (DHT11_PORT, DHT11_PIN, 1); // poner el pin en 1
70     delay(30); // esperar 20us
71     Set_Pin_Input(DHT11_PORT, DHT11_PIN); // poner el pin como entrada
72 }
73
74 uint8_t DHT11_Check_Response (void){
75     uint8_t Response = 0;
76
77     delay(40);
78     if (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))){
79         delay (80);
80         if ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))){
81             Response = 1;
82             delay(40);
83         }
84     }
85     //while ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))); // esperar a que el pin vaya a bajo
86
87     return Response;
88 }
89
90 uint8_t DHT11_Read (void){
91     uint8_t i, j;
92
93     for (j=0; j < 8; j++){
94         while (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))); // esperar a que el pin vaya a alto
95         delay(40); // esperar 40 us
96         if (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))){ // si el pin es bajo
97             i &= ~(1 << (7 - j)); // escribir 0
98         }
99         else
100             i |= (1 << (7 - j)); // si el pin es alto, escribir 1
101         while ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))); // esperar a que el pin vaya a bajo
102     }
103     return i;
104 }
105
106 int main(void)
107 {
108     HAL_Init();
109
110     SystemClock_Config();
111
112     MX_GPIO_Init();
113     MX_I2C1_Init();
114     MX_TIM2_Init();
115
116     HAL_TIM_Base_Start(&htim2);
117     Lcd_Init();
118     Lcd_Gotoxy(1, 1);
119     Lcd_Print("INICIANDO>>>");
120     HAL_Delay(2000);
121     Lcd_Command(LIMPIARLCD);
122     HAL_Delay(10);
123     Lcd_Gotoxy(1, 1);
```

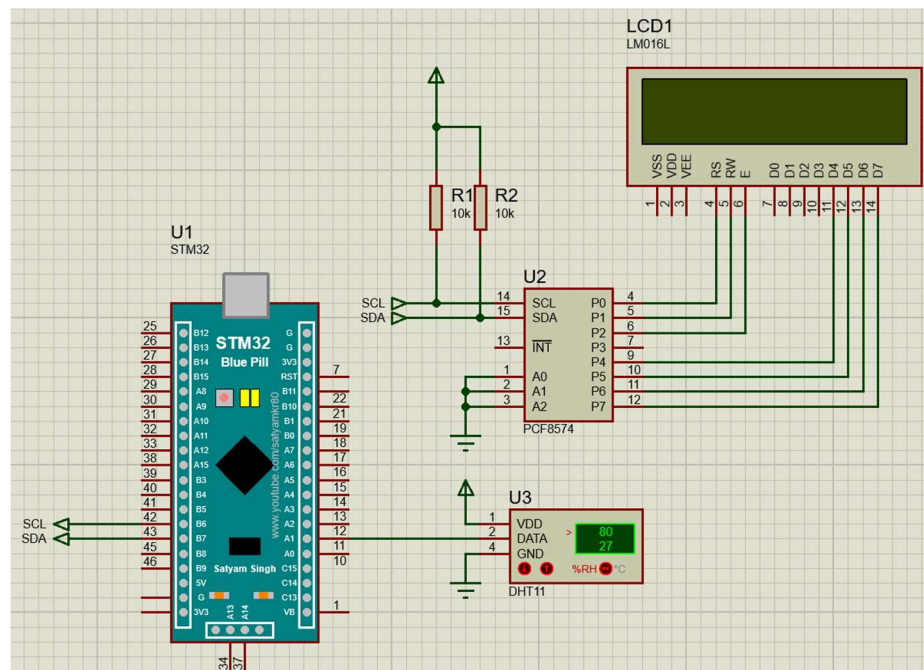
```

125 while (1)
126 {
127     Display_Temp(Temperature);
128     Display_Rh(Humidity);
129
130     DHT11_Start();
131     Presence = DHT11_Check_Response();
132     if(Presence){
133         Rh_byte1 = DHT11_Read ();
134         Rh_byte2 = DHT11_Read ();
135         Temp_byte1 = DHT11_Read ();
136         Temp_byte2 = DHT11_Read ();
137         SUM = DHT11_Read();
138     }
139
140     if(SUM == (Rh_byte1 + Rh_byte2 + Temp_byte1 + Temp_byte2)){
141         TEMP = Temp_byte1;
142         RH = Rh_byte1;
143
144         Temperature = (float) TEMP;
145         Humidity = (float) RH;
146     }
147
148     HAL_Delay(1000);
149 }
150 }
151 }

```

## 5. Esquemático del circuito.

El esquemático de la práctica se muestra a continuación.







## **6. Observaciones.**

Esta sección es para que el alumno anote sus observaciones.

## **7. Conclusiones.**

Esta sección es para que el alumno anote sus conclusiones.

## **8. Importante.**

La práctica deberá ser validada en el salón de clases antes de anexar el reporte al manual de prácticas. Una vez validada realizar el reporte de práctica como se anteriormente y anexar al manual de prácticas que se entregará a final del curso.