PROYECTO HIGRÓMETRO-TERMÓMETRO IoT

OBJETIVO

Creación de un dispositivo que mida temperatura y humedad y envíe los datos con una determinada frecuencia a un servidor Web accesible para el cliente.

CONCEPCIÓN INICIAL

\* Diseño basado en un sensor de humedad / temperatura de bao costo como el DHT 11 o el DHT 22

\* Comunicación Web a través del ESP 8266

\* Opcional: Monitoreo de estado de batería a través de ADC interno de microcontrolador ATMega 328P.

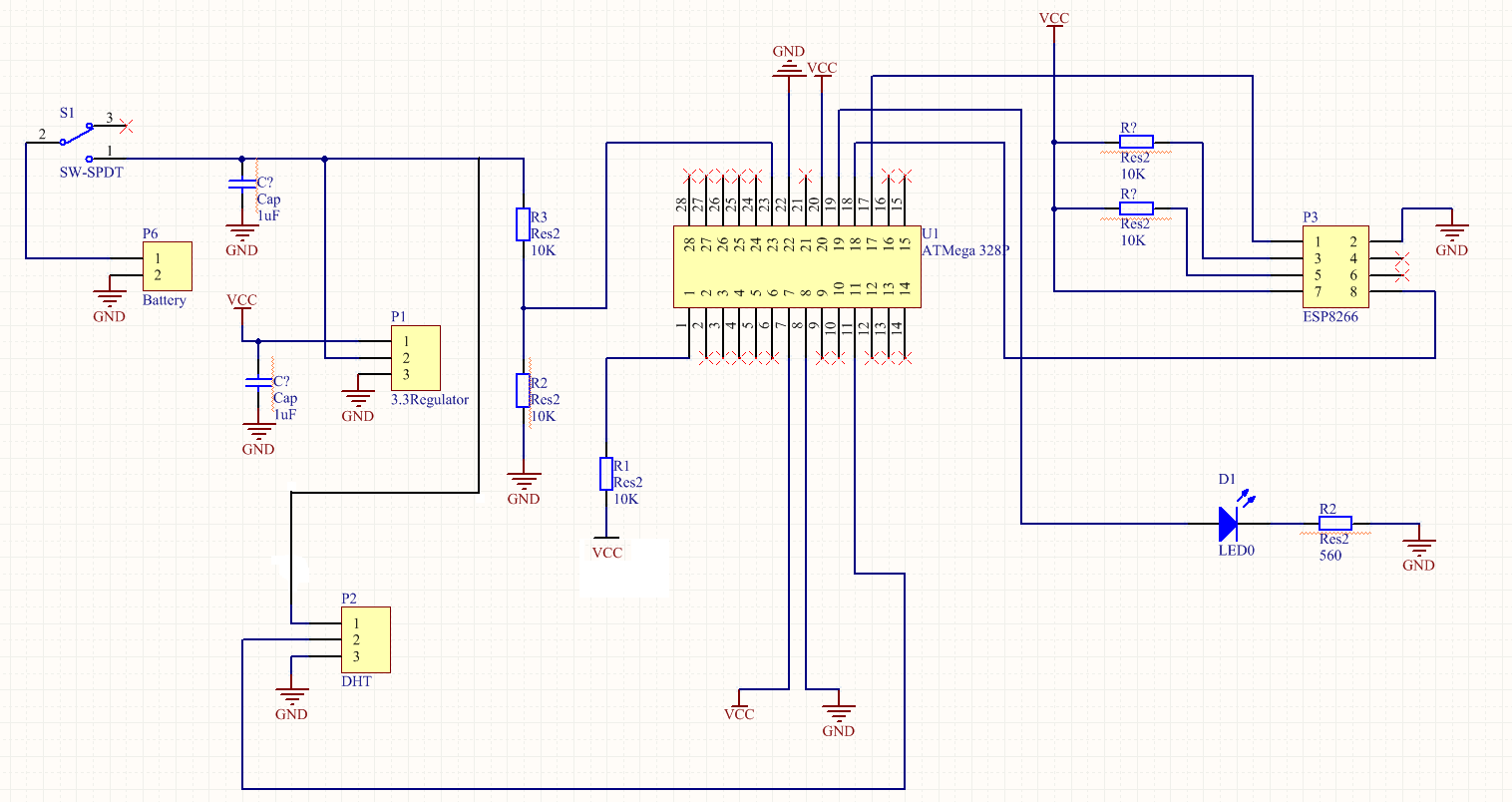
\* Alimentación: Baterías . Opción a) 1 de litio de 3.7 V, b) 3 alcalinas no recargables de 1.5 V, c) 4 de NiMH de 1.2 V.

\* Regulación: Usar un regulador a 3.3 V de baja pérdida por las características exigentes del ESP 8266.

DESCRIPCIÓN DEL PROCESO

- Diseño del circuito:

El circuito propuesto se presenta a continuación:

(ver en tamaño real en imagen adjunta “design.png”).

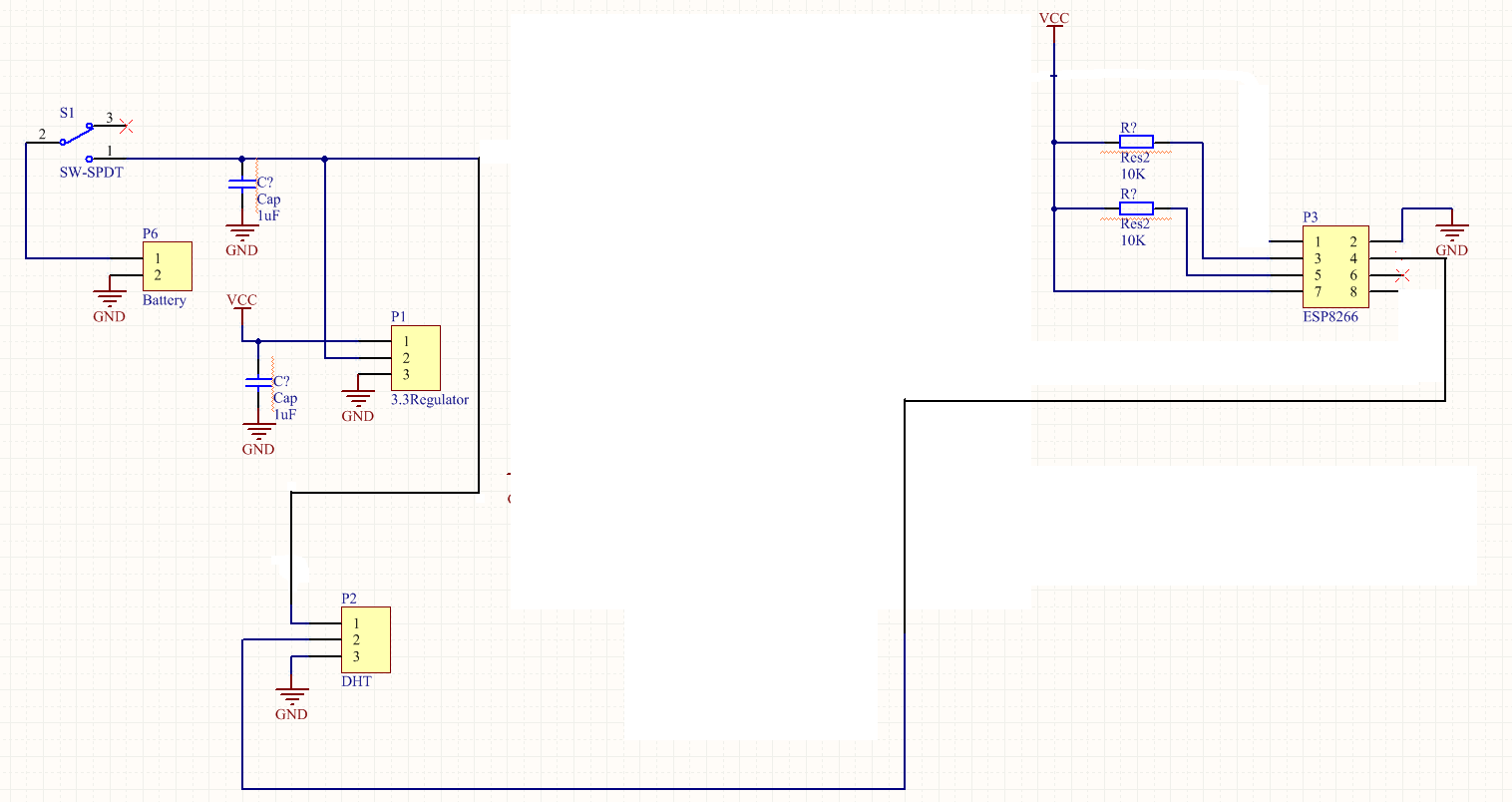
1. Entrada de alimentación de batería controlada por un interruptor.

2. Etapa de regulación: La tensión de entrada debe ser >= 3.4 V siempre para garantizar la salida de 3.3V del regulador de baja pérdida.

3. La tensión directa de la batería (sin regular) se pasa por un divisor de voltaje para poder utilizar el ADC del micro para leer nivel de batería. También se recomienda alimentar directamente de la batería el sensor DHT.

4. El micro debe tener una resistencia de reset (pin 1) a 3.3V. Hay un Led que se enciende cuando inicia el programa, pero se apaga de inmediato para no gastar energía. Se enciende también para indicar estado bajo de la batería.

5. El ESP8266 tiene dos resistencias de pull-up a 3.3V (chip-enable y reset). Se comunica por puerto serial con el micro.



( Versión alterna)

(ver en tamaño real en imagen adjunta “design-alt.png”).

Concepto similar al anterior, pero sin micro (sin indicación de batería y sin led). La conexión del sensor es al pin GPIO02 del ESP8266.

- Hardware

Ver adjunto “Materiales\_Higro.xlsx” para lista de componentes, precios, cantidades y lugares dónde conseguir.

- Software

Se requiere programar por aparte tanto el ESP-8266 como el ATMega 328P. Para ambos se utiliza la IDE de Arduino, disponible para Windows, MacOs o Linux en su página oficial:

<https://www.arduino.cc/en/main/software>

Aparte de eso, cada uno tiene su propio hardware de programación y un programa diferente. Vamos a ver cada uno de ellos.

\* ESP 8266

Acá un excelente tutorial que explica el circuito a montar para la programación y un ejemplo básico de programa (que se puede utilizar para probar los ESP al hacerles la modificación necesaria).

<https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>

Hardware: Circuito simple ya entregado a Gustavo, pero se recomienda leer el apartado en el tutorial para entenderlo.

Software: Seguir el mismo tutorial para descargar las librerías adicionales necesarias.

Ver archivo adjunto “ESP\_Tx\_Hum\_Temp.ino” para el programa a cargar (versión completa del circuito). Para versión alterna (sin micro), ver este programa: “DHT11\_ESP8266.ino”. En cualquier caso, no olvidar también las librerías del sensor DHT, disponibles donde indica este tutorial:

<https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

\* ATMega 328P

En este buen tutorial se explican cosas relativas a su programación utilizando un Arduino Nano como ISP (In-System Programmer)

<http://www.martyncurrey.com/arduino-nano-as-an-isp-programmer/>

Hardware: Circuito simple ya entregado a Gustavo, pero se recomienda leer el tutorial completo para entenderlo. Ojo con la parte de la frecuencia, por defecto viene a 1 Mhz (NO es cierto, viene a 8 Mhz según datasheet, pag.30), por lo que el man en el tutorial la modifica a 16 Mhz, lo cual requiere cristal externo. En nuestra aplicación trabajamos a 8 Mhz para prescindir del cristal, por lo que esa parte (la que habla del “boards.txt file”) no debe hacerse. Acá el datasheet del micro por si algo:

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>

Software: Ver archivo adjunto “Micro\_ADCtoESPserial.ino” donde está el programa a cargar en el micro. Ojo, hay una librería adicional para el modo de bajo consumo, que se llama JeeLib, en este tutorial se explica por qué se recomienda esa y de dónde bajarla:

<https://openhomeautomation.net/arduino-battery>

DESARROLLO WEB

(En construcción...pŕoximamente avances)