Universidad Nacional Experimental del Táchira
Código 0236509T - Automatización
Ingeniería Informática

Informe Técnico

Recuperativo

Asunto: Industria de petróleo

Integrantes:

- Baca Angelica C.I. V-24780105
- Valladares Yuleidy C.I. V-24147599
- Zambrano Loreana C.I V-24356109
- Zambrano Maribal C.I V-24.819.737
- Reyes Jesús Alfredo C.I V-24.152.665
- Vecino Yosswan C.I V-25498799
- Ramirez Lusdey C.I. V-20899002

06/12/2016

Implementos usados

- 6 LEDS
  3 para salidas analógicas.
  3 para salidas digitales.
- 2 Potenciómetros, ambos para entradas analógicas
- 1 Sensor infrarrojo, para entrada analógica.
- Cable UTP.
- Protoboard.
- Arduino nano.
- NodeMCU.

Modbus

Modbus es un protocolo de comunicación serial publicado por Modicon (ahora Schneider Electric) en 1979 para ser usado en sus controladores lógicos programables (PLC). Su uso está ampliamente extendido en la industria, sobre todo en los Sistemas de Supervisión, Control y Adquisición de Datos (SCADA), y sirve para comunicar los sistemas de supervisión remotos con las unidades terminales remotas (RTU). Éste protocolo se halla en la capa 7 del Modelo OSI (nivel de aplicación).

Modbus ofrece al maestro de la conexión 4 tipos de dato provenientes del esclavo:

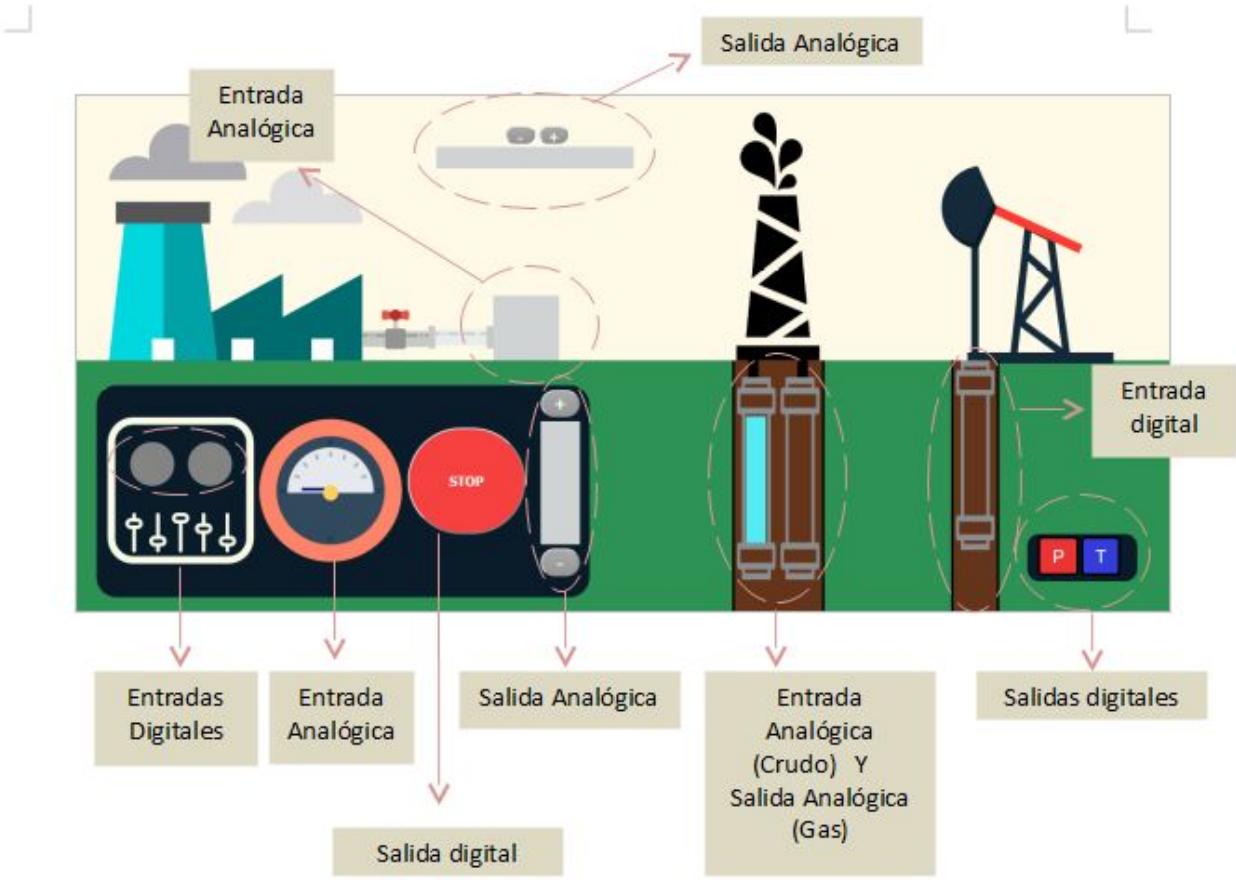
Tipo de objeto	Acceso	Tamaño (en bits)
Coil (bobina)	Lectura/Escritura	1
Dsicrete Input (entrada discreta)	Sólo lectura	1
Input Register (registro de entrada)	Sólo lectura	16
Holding Register (registro de almacenamiento)	Lectura/Escritura	16

La comunicación entre el maestro y el esclavo posee ciertas reglas:

- Cada dispositivo que se comunica usando Modbus posee una dirección única.
- El dispositivo maestro se comunica con los demás utilizando comandos de lectura/escritura (dependiendo del tipo de dispositivo configurado).

- Cada comando contiene la dirección Modbus del dispositivo al que va dirigido (las direcciones están en el rango de 1 a 247).
- Sólo el dispositivo al que se dirige el comando puede actuar ante éste, aunque los demás dispositivos de la red Modbus pueden recibirlo.
- El nodo 0 corresponde a la dirección de broadcast para todos los dispositivos.
- Sólo el dispositivo maestro puede iniciar un comando (si otro lo hace se genera una excepción).
- Todos los comandos Modbus contienen información de comprobación, para permitirle al receptor detectar errores de transmisión. Si el mensaje no se recibe correctamente se genera una excepción.

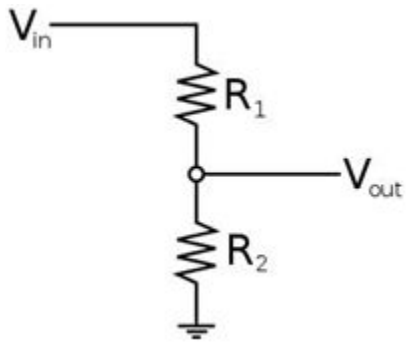
Proceso



Arduino Nano Y NodeMCU

Para la conexión entre Arduino y NodeMCU se realizó en la circuitería un divisor de voltaje, definiéndolo como un circuito simple que reparte la tensión de una fuente entre una o más impedancias conectadas. Con sólo dos resistencias en serie y un voltaje de entrada,

se puede obtener un voltaje de salida equivalente a una fracción del de entrada. Se usó debido a la diferencia de voltaje que sale del Arduino y la del NodeMCU siendo superior la del Arduino, que expulsa 5V y el NodeMCU soporta 3.3V, para realizar este divisor de voltaje se usaron 3 resistencias de 3.3 KΩ con el siguiente circuito detallado a continuación:



Para iniciar con el proceso se realizó la conexión entre Arduino, NodeMCU y página Web, siendo la comunicación NodeMCU-Página por el protocolo modbus que describimos anteriormente.

NodeMCU

Dentro del programa cargado en el NodeMCU se especifican una serie de pasos para la configuración y utilización del protocolo Modbus. Sirviendo de puente entre la página y el arduino nano. Las cuales se explican a continuación:

1. Se importan las librerías que permiten al ESP8266 conectarse a través de la red WiFi.

2. Se importan las librerías que implementan el protocolo Modbus dentro del programa de control.

3. Se importan las librería para la conexión en serie con el arduino nano (SoftwareSerial).

4. Se crea un objeto de tipo **ModbusIP** llamado mb, un objeto tipo **ESP8266WebServer** que manejará la pagina que esta dentro del nodemcu para configurar la red wifi a la que éste se va a conectar y un objeto tipo SoftwareSerial para la trasmisión de datos con el arduino donde se indica los pines para este tipo de comunicación RX GPIO13, TX GPIO15..

5. Se implementa la página para la configuración del WiFi a donde se va a conectar el NodeMCU, esta página se encuentra dentro del nodemcu. La red que se usa para la página web que configura la red es la que genera este.

String calcularSenal(long rssi) esta función se encarga de calcular la intensidad de la señal de las redes que tengo cerca.
void redes() esta función se encarga que imprime en la página la intensidad de cada red

6. Se conecta el NodeMCU a la red WiFi que se escogió en la página. Se espera por 10 seg a que la conexión sea exitosa. Pasado este tiempo se lanza un mensaje de error o de éxito, dependiendo del caso. Si es positiva se muestra la dirección ip que le asignará, el dispositivo que genera la red, al NodeMCU.

7. En **setup()** se configura el objeto modbus (mb) para que detecte la red a la que se conectará utilizando la función **mb.config("red","contraseña");** pasando por parámetros la red y contraseña que se escogió en la página web**.** Se espera por 10 seg a que la conexión se realice. Pasado este tiempo se lanza un mensaje de error o de éxito dependiendo el caso. Si es positiva imprimo la ip que le asignó, el dispositivo que genera la red, al NodeMCU.

8. Se configuran los coils y registros del objeto ModbusIP con la función **Mb.addCoil(5);** y **Mb.addHreg(6, 0);.** Las direcciones que se asignaron para los coils son de 0 a 5 y para los registros de 6 a 11.

9. En **void loop()** se valida la conexión a la red, si esta es negativa se muestra un mensaje de desconexión de lo contrario el objeto ModbusIP empieza a recibir y enviar comandos a través del protocolo Modbus con la función **mb.task();** Es importante acotar que se usó un delay de 200 ms antes y después de recibir y enviar data para la página y el arduino nano, esto es para poder sincronizar los tres elementos (página web, nodeMCU y arduino nano) de lo contrario se producían excepciones de expiración de tiempo de respuesta en la página web.

10. Se declara **String linea;** variable que contendrá los valores de los coils y registros y así con la ayuda de ciclos se mantendrán actualizados, mostrandose a su vez en la consola del IDE Arduino.

11. Los dos primeros ciclos se usan para guardar en la variable línea las salidas tanto digitales como analógicas. Se envían al arduino a través de **arduino.println(linea);**

12. Los dos últimos ciclos se usan para guardar en la variable linea las entradas tanto digitales como anagolicas. Se imprime todo en consola del IDE Arduino con **Serial.println(linea);**

13. Se usa la función **arduino.available()** para saber si hay bytes disponibles para leer desde el puerto serial. En el ciclo se lee lo que llega del arduino, Mb.Coil(var,arduino.parseInt()==1) para los coils y Mb.Hreg(var,arduino.parseInt()) para los registros.

Arduino Nano

1. Se definen los pines analógicos dentro de un array (0,1,2,3,5,6) identificándolos respectivamente en otro array llamado pines_analogicos_funcion, siendo 0 entradas y 1 salidas, de igual manera se hace para lo pines digitales (7,8,9,10,11,12), identificándolos respectivamente en otro array llamado pines_digitales_funcion, entradas 0, salidas 1.

2. En la función **leerEntradasAD()** se leen los valores desde el arduino nano y se almacenan en un vector llamado valores_digitales o valores_analogicos dependiendo del caso.

3. La función **cambiarEstadoDigital(int i, int estado)** recibe como parámetros el pin (digital) y el valor que llega desde el nodeMCU y se manda a activar o desactivar dicha salida, según sea el caso.

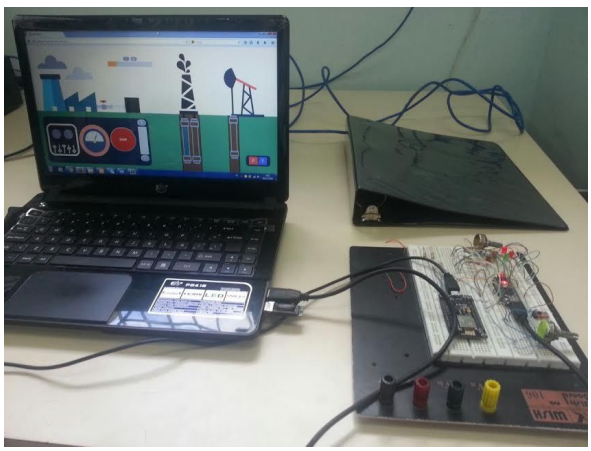
4. La función **escribirSalidaAnalogica(int i, int valor)** recibe como parámetros el pin (analógico) y el valor que llega desde el nodeMCU, el valor se restringe entre 0 y 255 y se manda a escribir.

5. En la función Setup() se declaran los pines de entrada y salida descritos en el primer item.

6. En la función loop() se verifica si llegó información desde el nodeMCU, se toma el pin y el valor que debe reflejarse en el arduino nano. Luego se verifica si se trata de un pin analógico o digital y se llama a la función correspondiente **cambiarEstadoDigital(indice,value);** o **escribirSalidaAnalogica(indice,value);**

7. Cada 1000 ms envía los valores digitales y analógicos al nodeMCU.

8. Dentro del loop() se van escuchando las entradas digitales y analógicas.



Nota: recuperación.

Se monta la página en router mediante un pendrive, se repite el proceso.

Para esto se modifica el código des Nodemcu, para que él sea el que mande a la página valores digitales (esto se debe a la ausencia del arduino).

Se comenta el código donde el Nodemcu recibe las entradas digitales del arduino, ya que este segundo dispositivo no se encuentra en el montaje.

```
    Serial.println(linea); // imprimir en consola la variable linea

  /*  if(arduino.available()>0) // si hay bytes en el puerto y el numero de bytes disponibles para leer desde el puerto
    {
      for(int i=0;i<6;i++) // leo lo que me llega del arduino
      {
        int var = arduino.parseInt();
        if(var<6)
          Mb.Coil(var,arduino.parseInt()==1); // coils
        else
          Mb.Hreg(var,arduino.parseInt()); // registros
      }
      if(Mb.Hreg(8)>=900) // prueba, encender el led del nodencu dependiendo lo que me llegue en el registro 8
        digitalWrite(2,0);
      else
        digitalWrite(2,1);
  }*/
}
delay(200);
```

Código comentado en el .ino del Nodemcu

Se conecta al pin D3 del Nodemcu a un cable que hará de entrada digital, esta se mostrará en la página, conectar el cable a 3.3v se devuelve un 1 (true, se enciende un dispositivo en la página) y a tierra se devuelve un 0 (false, se apaga un dispositivo en la página).

Abajo del código comentado se agrega la linea que me escribe el valor de D3 en los coils, para esto uso los coils con dirección de memoria 0 y 1

```
    }*/
  }
  delay(200);

  digitalWrite(0,mb.Coil(0,digitalRead(D3)));

  digitalWrite(1,mb.Coil(1,digitalRead(D3)));

}
```

Estas últimas dos líneas me escriben el valor que se va a mostrar en la página. La comunicación Nodemcu con la página se describe arriba, se usa el protocolo modbus.