

MONITOR FLUX - SPRINT 3

Integrantes: Jhoer Sebastián Pérez Castillo - Luis Fernando León Carvajal

Para el desarrollo de este sprint se tenía previsto las pruebas de funcionamiento al hardware seleccionado, tal y como se evidenció en los sprint anteriores la comunicación se realizará mediante el protocolo GSM, por lo cual se seleccionó el módulo **SIM 800L**, el cual en conjunto con la tarjeta **Arduino Uno** nos permitirá gestionar las solicitudes realizadas en la APP que se creara para tal fin, la simulación de los sensores que monitorean el estado de funcionamiento de las electrobombas se simulará a través de pulsadores y el estado de encendido o apagado de las mismas se simulará con diodos Led.

En la Figura 1 se puede apreciar el montaje realizado para poder cumplir con el desarrollo de este Sprint, allí se observa el hardware mencionado anteriormente.

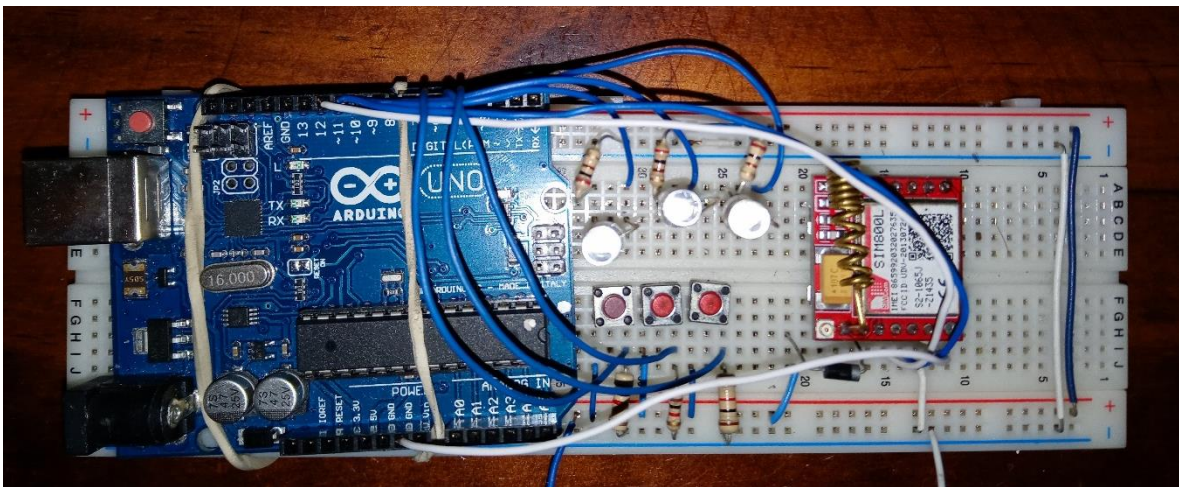


Figura 1. Hardware seleccionado implementado en una protoboard.

Una vez realizado el montaje en la protoboard se procedió a desarrollar el código, utilizando el IDE de arduino, allí fue posible configurar el módulo SIM 800L a través de comandos **AT** para que recibiera y enviara mensajes de texto, en la Figura 2 se puede apreciar el código implementado en arduino para tal fin.

```

#include <SoftwareSerial.h> //Se incluye esta libreria para poder crear otro puerto
//serial
SoftwareSerial mySerial(10,11); //Se crea otro puerto serial 10 - RX y 11 - TX
void setup() {
  //Se declaran los pines como entradas y salidas
  pinMode(6, INPUT);
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  //Se inicializa el puerto serial
  Serial.begin(9600);
  mySerial.begin(9600);
  //Se imprime un mensaje en el puerto serial
  Serial.println("Iniciando");

  mySerial.println("AT+CMGF=1"); //Configuramos el módulo en modo texto
  delay(1000); //Retardo de 1 segundo
  mySerial.println("AT+CMMSI=1,2,0,0,0"); //Se configura el módulo para ver la respuesta
  //en el puerto serial

}

void loop(){

  if(valor.indexOf("ON")>=0) //Si se recibe el mensaje ON se ejecuta una acción
  {
    digitalWrite(9,HIGH);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    delay(1000);
  }
  if(valor.indexOf("OFF")>=0) //Si se recibe el mensaje OFF se ejecuta una acción
  {
    digitalWrite(9,LOW);
    digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
    delay(1000);
  }
}

```

Figura 2. Código implementado en el IDE de arduino.

Una vez compilado y cargado el código se procedió a enviar los mensajes de texto desde el celular al módulo con las palabras ON y OFF. En la Figura 3 se pueden apreciar los mensajes de texto que se enviaron, en la Figura 4 se puede apreciar a través del puerto serial la fecha y hora en la que se recibieron los mensajes y en las Figura 5 y 6 se pueden observar las acciones que se programaron para cada mensaje, todo esto se puede corroborar con el código.



Figura 3. Mensajes enviados desde el celular.



Figura 4. Evidencia a través del puerto de serial de que se recibieron los mensajes.

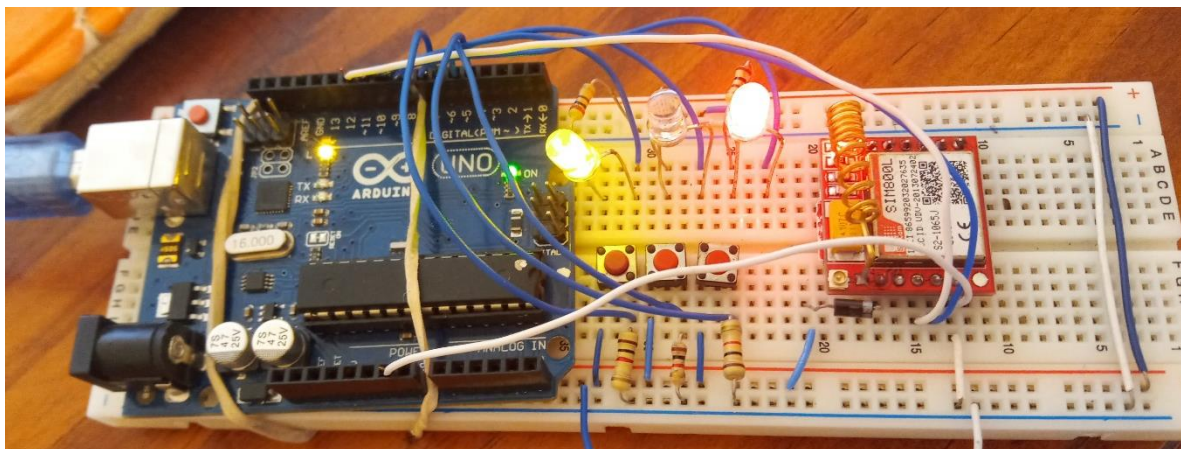


Figura 5. Acción programada para el mensaje “ON”.

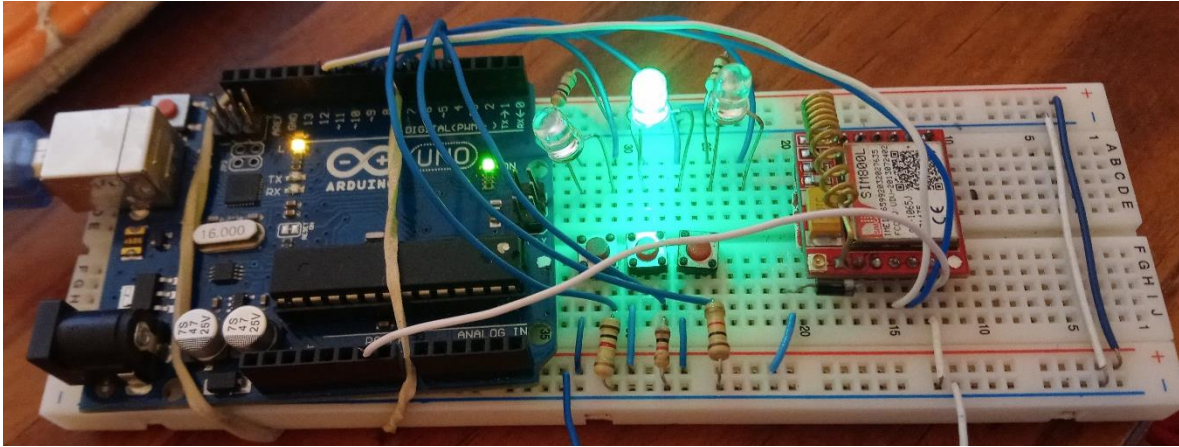


Figura 5. Acción programada para el mensaje "OFF".

Tal y como se evidencia en las imágenes anteriores el módulo ya está configurado para recibir mensajes y realizar acciones de acuerdo al texto que se reciba en los mensajes. También se realizaron pruebas para comprobar que el módulo también es capaz de enviar mensajes de texto; esto se programó para que cuando un pulsador se accionará (Falla en una electrobomba) el módulo envíe un mensaje. En la Figura 6 se puede observar el código que se añadió al código de la Figura 2 para tal fin.

```
if(digitalRead(6)==HIGH)
{
    mySerial.write("AT+CMGF=1\r\n"); //Se configura el módulo en modo texto
    delay(1000);                      //Retardo de un segundo
    mySerial.write("AT+CMGS=\"3142812968\"\r\n"); //Se establece el número al cual
                                                // se enviara el SMS
    delay(1000);                      //Retardo de un segundo
    mySerial.print("SIM800L - Recibido"); //Se escribe el mensaje
    delay(1000);                      //Retardo de un segundo
    mySerial.write((char)26);         //Se envia al módulo el comando ctrl z (char 26), para
                                    //indicar que el SMS termino
    delay(1000);                      //Retardo un segundo
}
```

Figura 6. Código implementado en Arduino para enviar un SMS.

En la figura 7 se puede apreciar el pantallazo del celular en el cual se recibió el mensaje.

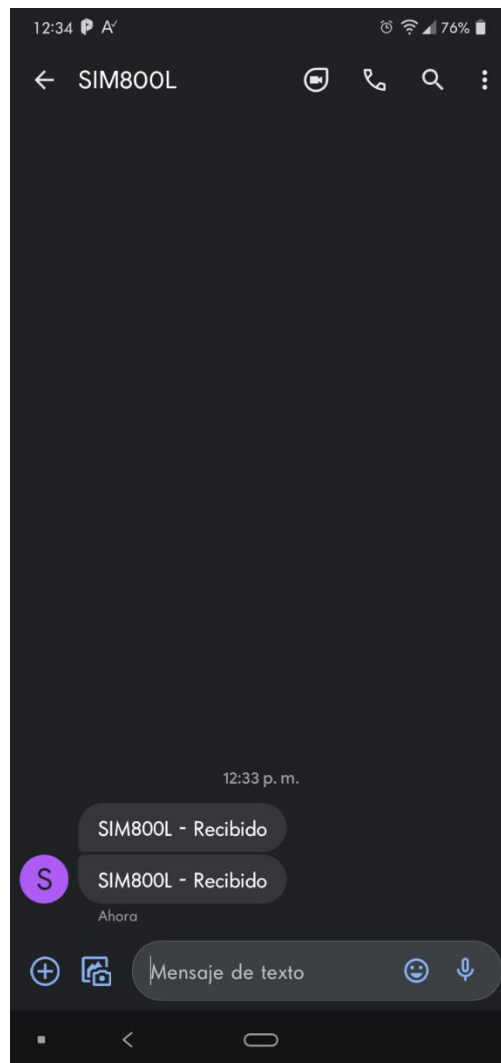


Figura 7. Mensaje recibido en el celular.

De esta manera se puede evidenciar que la integración del hardware fue buena y que las pruebas de funcionamiento dieron resultados positivos, pues el módulo junto con el arduino ya están en la capacidad de generar acciones específicas una vez se reciba un SMS y de enviar alertas a través de un SMS cuando se detecten eventos como el accionamiento de un pulsador (Falla en electrobomba). En los Sprints siguientes se comenzará a integrar el hardware con los avances que se llevan en la App.