Anexos

Especificaciones técnicas y programas de pruebas para Arduino

Autor

Daniel Mengod Simón

Director/es

Raúl Igual Catalán

Carlos Medrano Sánchez

Universidad Politécnica de Teruel

2019

1. Especificaciones técnicas

(a) SIM800L

Las características detalladas del componente son:

- *Voltaje de Operación: 3.4V - 4.4V DC*
- *Nivel Lógico de 3V a 5V*
- *Consumo de corriente (máx.): 500 mA*
- *Consumo de corriente (modo de reposo): 0.7 mA*
- *Interfaz: Serial UART*
- *Quad-band 850/900/1800/1900MHz – se conectan a cualquier red mundial GSM con cualquier SIM 2G*
- *Trabaja solo con tecnología 2G*
- *Capaz de enviar y recibir mensajes SMS*
- *Capaz de enviar y recibir datos GPRS (TCP/IP, HTTP, etc.) con una velocidad máxima de transmisión de 85.6 Kbps*
- *Escanear y recibir emisiones de radio FM*
- *Controlado por Comandos AT*
- *Interfaz de comandos AT con detección "automática" de velocidad de transmisión*
- *Velocidades de transmisión serial desde 1200bps hasta 115 200 bps*
- *Tamaño de la SIM: Micro SIM*

(b) ArduCAM mini 5mp plus

Las características detalladas del componente son:

- *Sensor de imagen de 5MP OV5642*
- *Lente reemplazable de montaje M12 / CS*
- *Interfaz I2C para la configuración del sensor.*
- *Interfaz SPI para comandos de cámara y flujo de datos*
- *Soporte de entrada de hardware externo.*

- *Admite el modo de compresión JPEG, modo de disparo único y múltiple, grabación de películas cortas, operación de lectura múltiple y operación de lectura de ráfaga.*
- *Todos los puertos IO son tolerantes a 5V / 3.3V*
- *Bien acoplado con tablas Arduino estándar*
- *Posee de una biblioteca de código fuente abierto para Arduino, ESP8266, Raspberry Pi, etc.*
- *Fuente de alimentación 3.3V ~ 5V*
- *Velocidad máxima de bus SPI: 8MHz*
- *Obturador: persiana enrollable*
- *Tamaño de píxel: 1.4 µm x 1.4 µm*
- *Tamaño: 34 x 24 mm 5MP, 1080p, 720p, VGA, QVGA*
- *Peso: 20 g*
- *Soporte de formato: RAW, YUV, RGB, JPEG*
- *Rango de temperatura: -10 ºC a 55ºC*

(c) Módulo SD

Las características detalladas del componente son:

- *Voltaje de Operación: 3.3V-5V*
- *Interfaz: SPI*
- *La tarjeta SD cuenta con todos los pines SPI: MOSI, MISO, SCK, CS*

(d) Panel solar

Las características detalladas del componente son:

- *Tensión de trabajo: 6 V*
- *Tensión de circuito abierto: 7.2 V*
- *Corriente de trabajo: 0 ~ 500 mA máx.*
- *Potencia: 3 W*
- *Eficiencia: 15 ~ 17%*
- *Dimensión: 145*145mm*
- *Peso: 0.095 kg*

(e) Regulador solar

Las características detalladas del componente son:

- *Desconexión de salida*
- *Protección contra cortocircuitos*
- *3W de potencia de salida al conectar la batería*
- *Corriente de carga continua hasta 900mA*
- *Indicación del estado de la batería (Rojo: Cargando, Verde: Cargado)*
- *Conector Micro-USB*
- *Especificación*
- *Voltaje de entrada de la batería: 3.0 ~ 4.5V*
- *Voltaje de entrada USB: 4.75 ~ 5.25V*
- *Voltaje de entrada solar: 4.8 ~ 6V*
- *Potencia máxima de salida (con batería): 3W (600mA a 5V)*
- *Tensión de ondulación: <100mV a 500mA*
- *Dimensiones: 68 * 53mm*

(f) Batería LiPo

Las características detalladas del componente son:

- *Tensión: 3.7V*
- *Corriente: 2000mAh*
- *Dimensiones: 60mm (largo) x 50mm (ancho) x 5mm (espesor)*
- *Peso: 36g*
- *Descarga: Menos de 8% por mes*
- *Rango de temperatura: -25 a 60C*

2. Programas de pruebas para Arduino

A continuación, se facilitan los dos programas mencionados en el capítulo del prototipo desarrollado.

(a) Envío y respuesta del módulo frente a comandos AT

```
#include <SoftwareSerial.h>    // Incluimos la librería SoftwareSerial
SoftwareSerial mySerial(10, 11);//Declaramos los pines RX(10) y TX(11)

void setup(){
Serial.begin(9600);        // Iniciamos la comunicación serie
mySerial.begin(9600);      // Iniciamos una segunda comunicación serie
delay(1000);               // Pausa de 1 segundo
}

void loop(){
if (mySerial.available()){// Si la comunicación mySerial tiene datos
Serial.write(mySerial.read());// Los sacamos por el puerto serie
}

if (Serial.available()){   // Si la comunicación serie tiene datos
while(Serial.available()) {       // y mientras tenga datos
mySerial.write(Serial.read());  // Los mandamos al módulo SIM800L
}
mySerial.println();               // Enviamos un fin de linea
}
}
```

(b) Almacenamiento de un archivo en la tarjeta SD

```
//Almacenamiento de un archivo en la tarjeta SD
//Se incluye la librería del módulo SD <SD.h>
#include <SD.h>

File Archivo;
//Pin CS del módulo conectado al pin digital número 10
int CS_pin = 10;

void setup() {

  //Se establece comunicación con el monitor serial para la
comprobación de la carga de datos.
  Serial.begin(9600);

  //Se muestra por pantalla que se va a iniciar la comunicación
  Serial.print("Comenzando la comunicación con la tarjeta SD");

  //Se establece como salida el pin correspondiente a CS.
  pinMode(CS_pin, OUTPUT);

  //Se muestra por el monitor si la comunicación se ha establecido
correctamente o ha habido algún tipo de error.

  if (!SD.begin(CS_pin)) {

    Serial.println("Se ha producido un fallo al iniciar la
comunicación");
    return;
  }
  Serial.println("Se ha iniciado la comunicación correctamente");

  /* ESCRIBIENDO DATOS EN LA MEMORIA SD DE ARDUINO */

  //Se abre el documento sobre el que se va a leer y escribir.
  Archivo = SD.open("datos.txt", FILE_WRITE);

  //Se comprueba que el archivo se ha abierto correctamente y se
procede a escribir en él.
  if (Archivo) {

    //Se escribe información en el documento de texto datos.txt.
    Archivo.println("Prueba SPI");

    //Se cierra el archivo para almacenar los datos.
    Archivo.close();

    //Se muestra por el monitor que los datos se han almacenado
correctamente.
    Serial.println("Todos los datos fueron almacenados");
  }
```

```
  //En caso de que haya habido problemas abriendo datos.txt, se
muestra por pantalla.
  else {

    Serial.println("El archivo datos.txt no se abrió correctamente");
  }

  /* FIN DE LA ESCRITURA DE DATOS EN LA MEMORIA SD DE ARDUINO */

  /* LEYENDO DATOS EN LA MEMORIA SD DE ARDUINO */

  //Se vuelve a abrir el fichero, esta vez para leer los datos
escritos.
  Archivo = SD.open("datos.txt");

  //Si el archivo se ha abierto correctamente se muestran los datos.
  if (Archivo) {

    //Se muestra por el monitor que la información que va a aparecer
es la del archivo datos.txt.
    Serial.println("Información contenida en datos.txt: ");

    //Se implementa un bucle que recorrerá el archivo hasta que no
encuentre más información (Archivo.available()==FALSE).

    while (Archivo.available()) {

      //Se escribe la información que ha sido leída del archivo.
      Serial.write(Archivo.read());
    }

    //Si todo ha ido bien cierra el archivo para no perder datos.
    Archivo.close();
  }

  //En caso de que haya habido problemas abriendo datos.txt, se
muestra por pantalla.
  else {

    Serial.println("El archivo datos.txt no se abrió correctamente");
  }

}

void loop()
{
  //En este ejemplo el bucle loop() no realiza ninguna acción ya que
toda la información fue gestionada en el setup.

  //En caso de que se desee almacenar la información obtenida de algún
sensor, la escritura debería realizarse en el loop().
}
```