Hardware

Integrated Microbial Dyeing Kit

This system innovatively integrates light-controlled dye synthesis, real-time feedback monitoring, dyeing process control, and remote communication functions, combining biological dye synthesis with intelligent control technology. It provides a novel solution for dye production, experimental research, and related industries. While the illustrations on our wiki have outlined the basic functions and applications of each module, some may require a deeper understanding of the complex technical details involved in the hardware design and implementation. This section will further elaborate and refine these technical details, particularly in key areas such as LED array control in the light-controlled dye synthesis module, sensor configuration in the feedback module, peristaltic pump control and dye delivery in the dyeing module, and 4G communication technology in the remote control module. These contents aim to help readers with a hardware background gain a better understanding of the system's hardware implementation, performance optimization methods, and technical challenges.

CONTENT

1. Light-Controlled Dye Synthesis Module

1.1 Features of WS2812 LED Light Array

Our curtain system uses a 16×16 LED array composed of 256 SK5812-021 LED beads. The SK5812-021 is an intelligent externally controlled LED light source that integrates both control circuitry and light-emitting circuitry. Its appearance is identical to that of a top-emitting SMD5050 LED bead, with each component serving as an individual pixel.

Each pixel contains an intelligent digital interface, data latch, signal shaping and amplification driver circuit, power regulation circuit, built-in constant current driver, and a high-precision RC oscillator. The output driver employs patented PWM technology, ensuring high color consistency within each pixel.

1.1.1 Pin Description

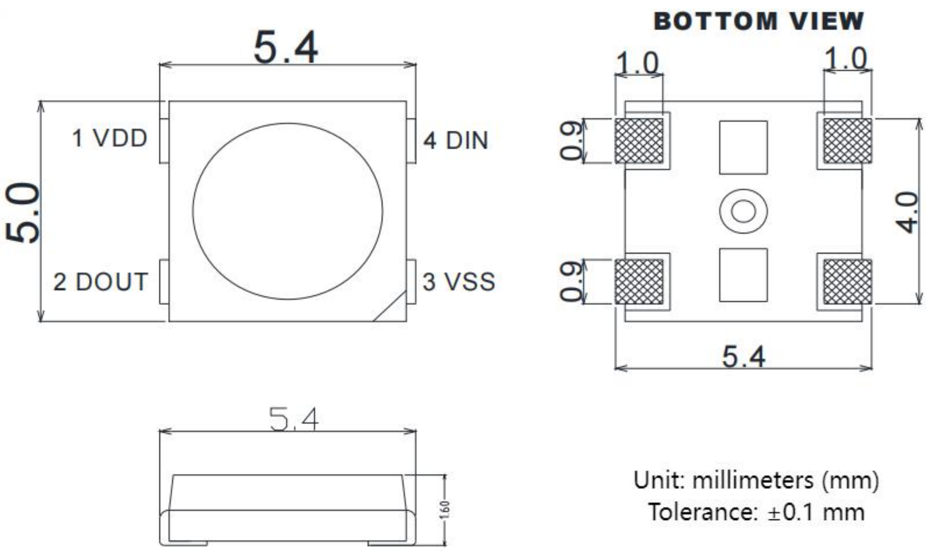


Figure 1 Appearance and Mechanical Dimensions of One Bead

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)

The pin layout is of one bead shown in the figure above, and the pin functions are described in the table below.The data protocol adopts a single-wire return-to-zero communication method. After power-on reset, the **DIN** pin of each pixel receives data transmitted from the controller. The first 24-bit data segment is extracted by the first pixel and stored in its internal data latch. The remaining data are reshaped and amplified internally, then output through the **DOUT** pin to the next cascaded pixel.

Table 1 Pin Function of One Bead

No.	Symbol	Pin Name	Function
1	VDD	Power	Power supply pin
2	DOUT	Data Output	Signal data output
3	VSS	Ground	Power ground
4	DIN	Data Input	Signal data input

1.1.2 LED Serial and Data Transmission

With each transmission through a pixel, the signal is reduced by 24 bits. The pixel's automatic reshaping and forwarding technology ensures that the number of cascaded pixels is not limited by signal transmission distance, but only by transmission speed requirements, which is suitable for our light-controlled dye synthesis project.

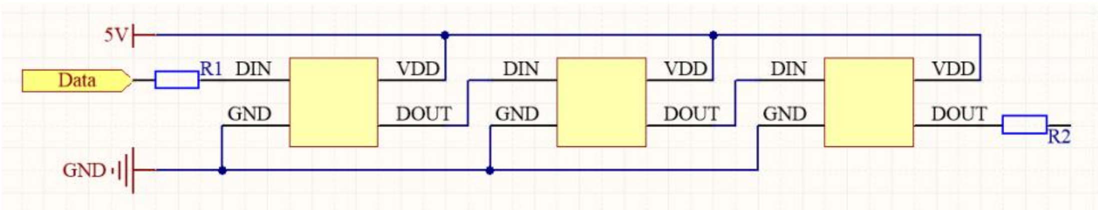


Figure 2 Typical Circuit

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)

This is a typical application circuit for LED series connection and signal transmission. In the actual circuit, protective resistors and decoupling capacitors should be added to protect the circuit and improve its stability.

When transmitting data, the 24-bit data structure is as below. High bit data will be sent first. And data are transmitted in the order of GRB (Just as the figure below: G7 → G6 → … → B0).

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4
R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

Figure 3 Order of Transmission

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)

1.1.3 Feature Analysis

The LED light array composed of SK5812-021 LED beads has its features suitable for our project, which needs the precise regulation of the light intensity of three specific wavelengths within a defined region and the realization of mixed-color light.

- ❖ **Integrated Control Circuit**: The control circuit and LED chip are combined in an SMD5050 package, forming a complete externally controlled pixel with uniform color temperature and high consistency.
- ❖ **Gray-Scale Adjustment:** Supports 256 levels of brightness adjustment, which is the reason that the parameter setting range on our GUI interface is 0 - 255.
- ❖ **Optimized Red-Light Drive Processing:** Improves color balance and consistency in mixed-color applications.
- ❖ **Single-Wire Data Transmission:** Supports unlimited cascading of pixels through serial data transfer.
- ❖ **Long Distance Transmission:** Enhanced Signal Transmission Technology ensures stable data communication over distances exceeding 10 meters between points.
- ❖ **High Transmission Speed**: Supports data transmission rates up to 800 kbps; when the refresh rate is 30 frames per second, the cascade length can reach at least 1024 pixels.

1.2 Control Mode

Each LED pixel is governed by an independent “PixelTimer”, which uses a unified time reference to drive light modulation according to two waveform modes, boxcar/B (rectangular pulse) and sinusoi/S (sinusoidal interpolation). This architecture is suitable for applications requiring asynchronous pixel modulation, long-term timing stability, and precise rhythm control, especially for photobiological experiments.

1.2.1 PixelTimer Mechanism

Each pixel i is associated with a dedicated PixelTimer that stores:

$$color\ 1 = (r, g, b),$$

$$color2 = (r2, g2, b2),$$

$$timing\ parameters = (param1, param2),$$

$$\text{mode} \in \{\text{boxcar, sine}\}.$$

The runtime process is as follows:

- ❖ Obtain the global timestamp t (RTC-corrected milliseconds).
- ❖ Call process_time(t) for each pixel:

Boxcar Mode: Within one cycle $T = Ton + Toff$, if $t \leq Ton$, the LED displays color1; otherwise color2—implementing a discrete duty-cycle modulation.

Sine (sinusoi on our GUI) Mode: Uses $s(t) = \sin^2(2\pi t/T)$ as a normalized weight to interpolate smoothly between color1 and color2, producing a breathing or fading effect.

```
void PixelTimer::boxcar(unsigned long t){
  unsigned long w = t % (param1_ + param2_);
  if (w <= param1_) led_matrix.setPixelColor(my_pixel_, intensity_r_, intensity_g_, intensity_b_);
  else              led_matrix.setPixelColor(my_pixel_, intensity_r2_, intensity_g2_, intensity_b2_);
}

void PixelTimer::sine(unsigned long t){
  unsigned long w = t % param1_;
  double s = pow(sin(2.0 * PI * w / param1_), 2);
  led_matrix.setPixelColor(my_pixel_,
    s*intensity_r_  + (1-s)*intensity_r2_,
    s*intensity_g_  + (1-s)*intensity_g2_,
    s*intensity_b_  + (1-s)*intensity_b2_);
}
```

Figure 4 Code for Two Modes

Obviously, in our project we usually choose boxcar mode as our control mode based on the request from experiments. But we retained sine mode to enhance the general applicability of the program.

- ❖ Finally, a unified led_matrix.show() transmits the frame buffer, ensuring synchronization and efficiency.

1.2.2 Real Time Clock (RTC)

Because millis() (our built-in function that returns the amount of time that has elapsed since the system started running) relies on the MCU's internal oscillator, its timing drifts gradually during long operation, causing animation desynchronization. The DS3231 RTC provides a precise, temperature-compensated clock reference. Every 10 seconds, the firmware recalibrates time using the RTC's value.

The synchronization process:

- ❖ Compute elapsed seconds curr_sec = RTC_now - start_sec;
- ❖ When curr_sec exceeds a multiple of 10 s, reset start_millis;
- ❖ Apply offset correction to curr_millis for linear time alignment.

The code is as follows:

```
#if USE_RTC
  DateTime now = rtc.now();
  long curr_sec = now.secondstime() - start_sec;
  if (curr_sec - time_correct_interval >= prev_sec) {
    prev_sec = curr_sec;
    start_millis = curr_millis;
  }
  curr_millis += 1000 * time_correct_interval * floor(curr_sec / time_correct_interval);
  curr_millis -= start_millis;
#endif
```

Figure 5 Code for RTC

This mechanism ensures long-term temporal coherence, maintaining consistent lighting rhythm over hours or even days.

2. State Monitoring and Feedback Module

2.1 Features of GY-302 and PCA9548A

2.1.1 GY-302

The GY-302 uses the original **BH1750FVI** chip from **ROHM** and features five pins: **VCC**, **GND**, **SCL**, **SDA**, and **ADDR**. This chip features a spectral response closely matching human visual sensitivity and can output light intensity data ranging from 0 to 65,535 lux.

No.	Symbol	Function
1	VCC	Power supply pin
2	GND	Power ground
3	SCL	I2C communication
4	SDA	I2C communication
5	ADDR	Set address

Figure 6 Appearance and Pin Function of GY-302

The circuit schematic is shown below:

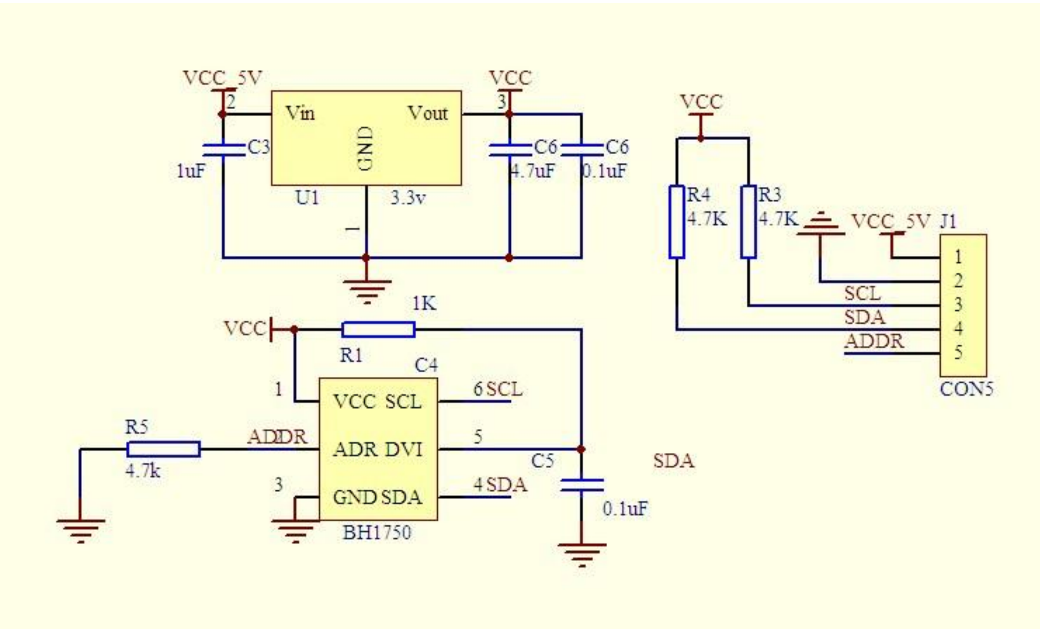


Figure 7 Circuit Schematic of GY-302

Its design concept is to convert the external 5 V power supply into 3.3 V through the voltage regulator chip U1 (Low Dropout Regulator), providing power to the sensor and the I2C bus. R3 and R4 serve as pull-up resistors for the I2C bus, ensuring stable communication of the SCL/SDA signals. R5 pulls the ADDR pin low to ground, setting the device's I2C address to the low address (0x23). R1 (1 kΩ) is used to suppress interference and signal reflections on the clock line. C4 and C5 are local decoupling capacitors for the sensor, improving power stability. The J1 connector provides the 5 V power supply, I2C signals, and address pin connection to the main control board. The overall circuit is well-structured, ensuring stable power supply and reliable I2C communication for the BH1750 sensor.

Therefore, we supply the GY-302 with a stable 3.3 V power source and properly connect the I2C communication interfaces SDA and SCL, allowing stable control of the BH1750 chip to read light intensity and obtain real-time illumination data.

2.1.2 PCA9548A

PCA9548A is a low-voltage, resettable 8-channel I2C multiplexer. A single upstream **SCL/SDA** pair can be selectively connected to any of the eight downstream channels (**SC0**/**SD0**…**SC7**), effectively resolving slave address conflicts and reducing bus capacitance. The device provides three address pins (**A2**,**A1**,**A0**) for parallel configuration of multiple chips and includes both a **RESET** (active low) input and a Power-On Reset (**POR**) function to clear the state machine and return the device to its default state in case of bus errors. It supports communication speeds of 100 kHz and 400 kHz, operates with a supply voltage of 2.3–5.5 V, and features 5 V-tolerant inputs.

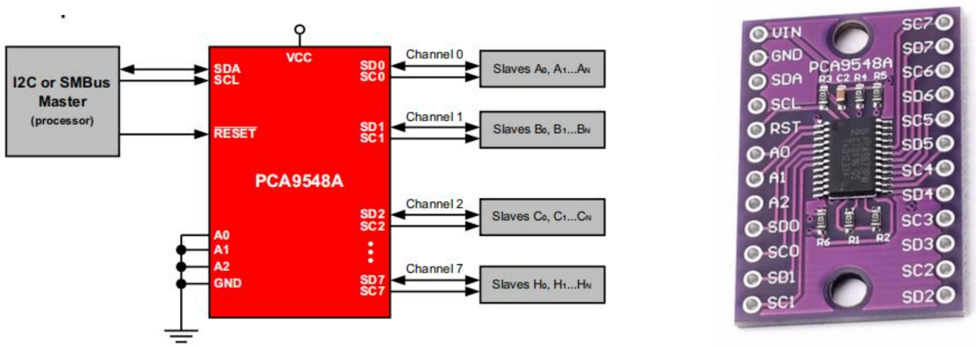


Figure 8 Schematic and Appearance (Provided by TEXAS INSTRUMENTS)

2.2 Communication Mode

Our multiplexing logic follows a 1–4–25 configuration. Specifically, one Arduino Mega 2560 mainboard's I2C communication interface is connected in parallel to four PCA9548A multiplexing modules by 1-to-4 Dupont cable. Each PCA9548A module can connect up to eight GY-302 I2C communication interfaces.

In total, the four PCA9548A modules connect to our 25 GY-302 sensors—the first three PCA9548A modules each connect to eight sensors, while the fourth module connects to a single sensor. Set the addresses of the four PCA9548A modules (from A2 to A0) to 000 (0x70), 001 (0x71), 010 (0x72), and 011 (0x73), respectively, so that the four modules can be individually distinguished.

This design effectively expands the single I2C interface of the Arduino to accommodate all 25 sensors simultaneously, achieving structured and efficient communication management across multiple sensor channels.

We have drawn the schematic diagram of feedback module using ***Proteus 8 Professional*** as below.

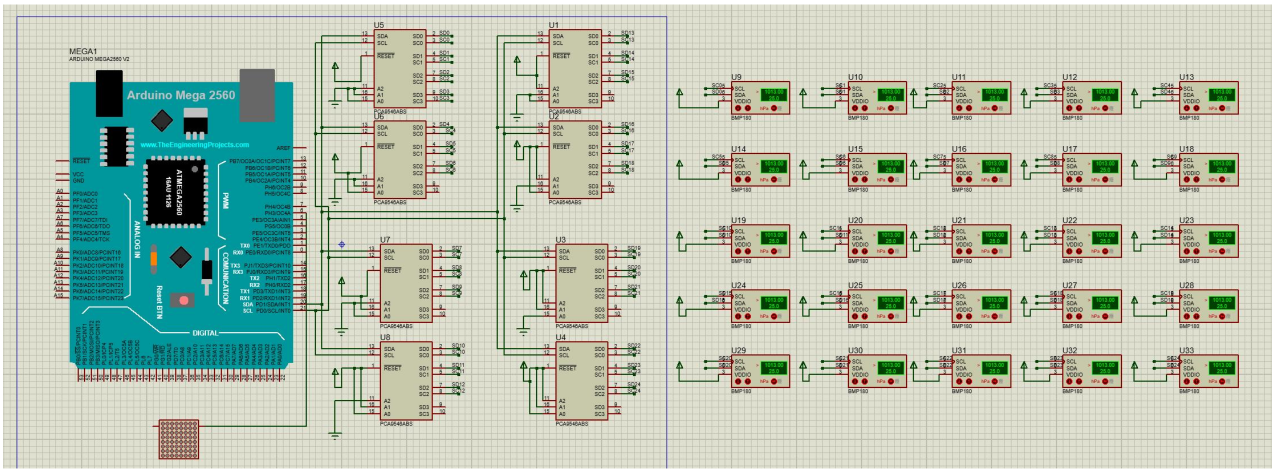


Figure 9 Schematic Diagram

In the lower-left corner is the LED array. Due to library limitations, an 8×8 LED array was used instead of the 16×16 one. The data input pin of the LED array is connected to pin 6 of the mainboard. The other pin, which serves as the data output pin required for LED array cascading, is not connected to any other component in this setup.

Due to software library limitations, we replaced the PCA9548A module (an 8-channel I2C multiplexer) with the PCA9546 module (a 4-channel I2C multiplexer) and substituted the GY-302 (a light intensity sensor using I2C communication) with the BMP180 (a barometric pressure and temperature sensor also using I2C communication).

This substitution does not affect the demonstration of the wiring principle, as both the PCA9546 and BMP180 follow the same I2C communication protocol and wiring logic. The setup still effectively illustrates the multiplexing and signal routing principles of the system.

2.3 Feedback Design

The system uses millis() as its primary timeline and periodically corrects it with a DS3231 RTC to avoid long-term drift . Each frame, PixelTimer generates time-based waveforms (boxcar or sine fades) for all 256 NeoPixels. Readers can see Section 1.2 for details.

Then the firmware polls 25 BH1750 ambient-light sensors through PCA9548A I2C multiplexers: it selects one downstream channel at a time by writing a bit mask to the PCA9548A, reads the lux value, and compares it to a per-system threshold (MIN_LIGHT, default 50 lux in the example code). The code can be seen below.

```
// Poll 25 BH1750 sensors and perform "blackout override"

#define MIN_LIGHT 50
uint16_t lightLevels[25];

for (int i = 0; i < 25; i++) {
  selectPCA9548A(i);                          // Switch to the i-th sensor.
  lightLevels[i] = lightSensors[i].readLightLevel();
  if (lightLevels[i] < MIN_LIGHT) {           // intensity below MIN-LIGHT→ blackout override
    for (int k = 0; k < 4; k++) {
      int ledIndex = base_to_leds[i][k];
      led_matrix.setPixelColor(ledIndex, 0, 0, 0);
    }
  }
}
```

Figure 10 Code for Blackout Override

The display logic implements a write-after “blackout override”: render the animation first, then forcibly set the four LED beads mapped to that sensor’s area (via base_to_leds) to black if the measured light is below the threshold, and finally call show() to output the frame. This realizes a “pattern-by-time + gating-by-sensing” loop—global content is time-driven, while local visibility is decided by real-time illumination, saving power in dim environments that indicate dye synthesis completed.

```
// Timeline: first execute the pattern ,
//then apply sensor-based gating override, and finally refresh the display.
unsigned long curr_millis = millis();  // RTC correction applied periodically
for (int i = 0; i < num_neopixels; i++) {
  ptimers[i].process_time(curr_millis); // Render the animation first.
}
// “Blackout override” code – to be discussed below.
led_matrix.show();                      // Output a single frame.
```

Figure 11 Code for Display (Including RTC)

3. Dyeing Module

Regarding the dyeing module, we have only carried out preliminary theoretical design and conceptual planning, and we look forward to further development in the future.

3.1 Peristaltic Pump

3.1.1 Introduction to Peristaltic Pump

A peristaltic pump is a device that transports fluids by rolling and compressing a flexible tube. With its precise flow control, convenient operation, and contamination-free fluid delivery, it perfectly meets the requirements of the dyeing module for “dye extraction and quantitative delivery.” It ensures uniformity and reproducibility in dye application during the dyeing process, thereby achieving a “uniform, controllable, and high-precision” dyeing effect.

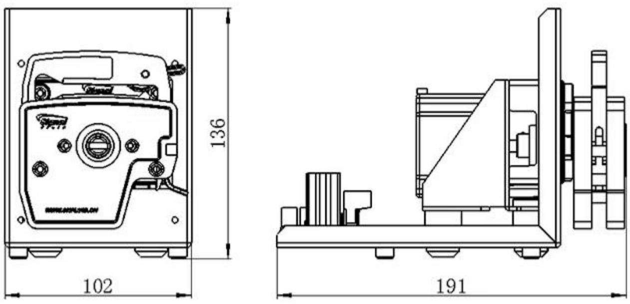


Figure 12 Appearance and Mechanical Dimensions of Common Peristaltic Pump

The pump head of the peristaltic pump consists of a “housing pressure block” and a “roller assembly”. The flexible tube is compressed between these two components along the working circle, forming a sealed chamber.

When the roller assembly rotates, the rollers sequentially compress the tube, pushing the dye inside forward toward the dyeing area. As the rollers move away, the tube’s elasticity allows it to return to its original shape, creating a vacuum that draws dye from the dye container.

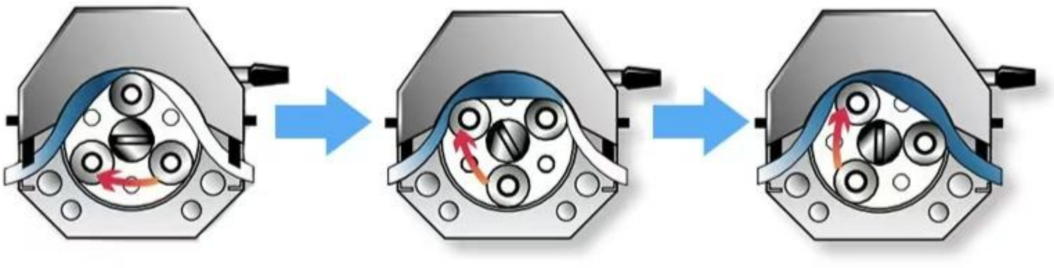


Figure 13 Working Principle of the Peristaltic Pump

The section of the tube between two rollers forms a “pillow-shaped dye column,” whose volume is determined by the inner diameter of the tube and the geometric dimensions of the rollers. The dye flow rate required for dyeing depends on the pump head rotation speed × pillow volume × number of pillows per rotation. Therefore, by adjusting the rotation speed, the dye delivery rate can be precisely controlled, enabling accurate regulation of the dyeing effect.

The reasons we choose peristaltic pump as our dye pump is as follows:

❖ **Precise Flow and Uniform Dyeing:** The system allows accurate control of dye extraction volume and delivery speed, preventing “uneven dyeing caused by excessive or insufficient dye”, thereby ensuring consistency in dyeing quality.

❖ **Contamination-Free Fluid Transfer and Purified Dye:** Only the inner wall of the flexible tube comes into direct contact with the dye, while the other components of the pump remain isolated from the fluid, which prevents the dye from contaminating the equipment and also avoids any potential effects.

- **Flexible Operation and High Adaptability:** The system supports multiple modes, including internal control (multi-level BCD dial switch) and external control (signals such as 0–10 kHz or 0–5 V, which we plan to use). This versatility allows it to be integrated into whether manual small-batch dyeing or automated dyeing production lines.

- **Adjustable Speed and Controllable Rhythm:** Driven by a stepper motor, the pump offers a speed range of 0–100 RPM with a resolution of 0.1 RPM, allowing flexible adjustment of dye delivery rhythm according to different dyeing requirements.

3.1.2 Our Plan

We plan to implement the external control mode of the peristaltic pump on the Arduino Mega 2560 platform, covering interfaces such as 0–10 kHz frequency input, 0–5 V and 0–10 V analog voltage, Start/Stop, and direction control. The project will proceed in three phases:

- **Step 1 — Signal Generation and Conditioning:**
  Using a hardware timer to generate a 0–10 kHz square wave (50% adjustable duty cycle), with an external MCP4725 DAC producing a true 0–5 V analog output. The signal is then amplified through an operational amplifier (gain = 2) powered by 12 V to achieve a 0–10 V range. For open-collector or optocoupler inputs, both NPN and optocoupler-isolated drive circuits with common-ground/isolated configurations will be provided. Basic EMI/ESD protection and RC/TVS component selection will be completed.

- **Step 2 — Pump-Side Integration and Calibration:**
  A mapping between control inputs, rotation speed, and flow rate will be established for the target peristaltic pump. Frequency–speed and voltage–speed curves will be derived, and flow rate calibration ($n \geq 5$ points per RPM) will be performed using the gravimetric or volumetric method. *Target KPIs: rotation speed error ≤ ±1%, volume error ≤ ±3% (1–50 mL range), and start/stop hysteresis < 0.5 s.*

- **Step 3 — Application and Safety:**
  The system will support quantitative volume control (target volume/target flow

rate modes), soft start/stop, emergency stop, interlocks for door access or low fluid level, and fault alarms (missed steps, overcurrent, disconnection). Optional 4–20 mA compatibility and closed-loop flow/pressure sensing interfaces will be reserved. *KPIs: fault cut-off < 100 ms and continuous operation MTBF ≥ 100 h.*

3.2 Nozzle Design

At present, the system adopts a valveless peristaltic pump direct-drive nozzle design, which enables stable dye dripping and delivery by controlling the pump's rotation speed and operation time. This approach features a simple structure, ease of implementation, and low maintenance cost, making it suitable for single-channel or low-channel liquid distribution.

However, as the system evolves toward multi-channel array configurations, limitations begin to emerge in terms of dripping precision, synchronization control, and anti-leakage performance. To enhance control accuracy and scalability, we are considering the integration of a micro solenoid valve array, allowing independent on/off control of each nozzle to ensure consistency and synchronization during multi-channel dyeing operations.

Based on the future lightweight design of our device, we will focus particularly on system lightweighting and wiring simplification. Traditional multi-valve systems often suffer from complex wiring and redundant control modules, hindering miniaturization and maintenance. To address this, we propose a design in which a single driver module controls a multi-channel solenoid valve array, using multi-channel MOSFET arrays or ULN2803-type driver chips to centralize valve control signals. This modular electrical structure optimizes wiring, reduces system weight and volume, and establishes a foundation for scalable multi-channel dyeing systems.

4. Remote Control Module

4.1 Features of E27V-S

The E27V-S is a 4G DTU product launched by Tastek LoT, belonging to the TAS-LTE-4G development board series. It is a pin-header-type device that integrates RF matching, power management, TTL level conversion, and LED status indicators, with its main functional pins exposed via header pins for easy insertion into user circuit boards. The product is also equipped with a 4G PCB antenna to enhance communication performance.

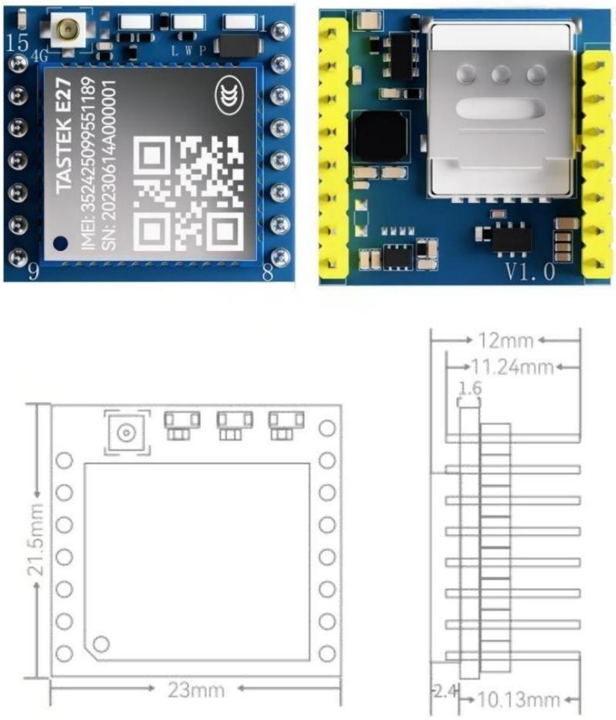


Figure 14 Appearance and Mechanical Dimensions of E27V-S

Users only need to provide a power supply interface for the device to operate. The module comes with comprehensive built-in software functions, covering the vast majority of standard application scenarios.

In our project, this module is used to achieve bidirectional transparent data transmission between the serial port and the network, thereby enabling remote control functionality.

4.1.1 Basic Feature

To meet the lightweight design requirements of our device, we specifically selected the compact E27V-S model from Tastek's 4G DTU product line. The E27V-S, powered by the EC716S main chip, operates on the LTE Cat 1 network standard. It can be configured via serial commands, host computer software, or network commands, and supports multiple network protocols including TCP, UDP, DNS, MQTT, and HTTP, enabling transparent data transmission.

The module features both embedded SIM (eSIM) and removable SIM card slots, operating in a dual-SIM single-standby mode with the embedded SIM taking priority. It is equipped with one IPEX connector for antenna connection and includes three 3.0–3.3 V TTL serial ports. The E27V-S operates reliably within a DC 5–16 V voltage range.

4.1.2 Pin Function

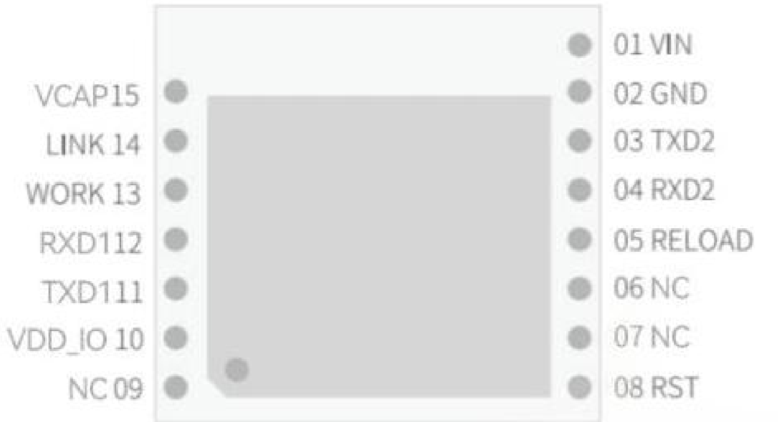


Figure 15 Pin Distribution

The pin layout is of E27V-S shown in the figure above, and the pin functions are described in Table 2 below. We primarily use the **VIN**, **GND**, **TXD1**, and **RXD1** pins — these are responsible for power supply and data transmission for the module. Since the module has a built-in work status indicator, we do not need to use pin-level outputs (output of **WORK** pin and **LINK** pin) to determine the operational status.Three built-in indicator lights are **P (Power)**, **W (Work)**, and **L (Link)** light. These lights indicate the following:

- ❖ **P (Power):** Indicate the power status. This light is always on when normal.
- ❖ **W (Work):** Indicate the working status. This light blinks when normal.
- ❖ **L (Link):** Indicate the socket connection status. It is always on when normal.

Table 2 Pin Function of E27V-S

No.	Symbol	Definition	Function
1	VIN	Power Input	Positive power supply input, 5 – 16 V. Leave unconnected if powered through the VCAP pin.
2	GND	Power/Signal Ground	Power supply ground.
3	TXD2	Serial Transmit	Data transmission from the module to the user device (supports 3 V – 3.3 V TTL levels).
4	RXD2	Serial Receive	Data received by the module from the user device (supports 3 V – 3.3 V TTL levels).
5	RELOAD	Factory Reset	Pull low for 3 – 15 s to restore factory settings (active low).
6	NC	Reserved	Leave unconnected.
7	NC	Reserved	Leave unconnected.
8	RST	Module Reset	Reset pin for the module; internally pulled up and active low. Leave unconnected if unused.
9	NC	Reserved	Leave unconnected.
10	VDD_IO	Reserved	Output voltage 3.3 V. Leave unconnected if unused.
11	TXD1	Serial Transmit	Data transmission from the module to the user device (supports 3 V – 3.3 V TTL levels).
12	RXD1	Serial Receive	Data received by the module from the user device (supports 3 V – 3.3 V TTL levels).
13	WORK	Work Indicator	Module operation indicator; flashes high once per second during normal operation.
14	LINK	Server Connection	Indicates server connection status: high level when connected, low when disconnected.
15	VCAP	3.8V Power Input	Connect an external 470 µF electrolytic capacitor or use as a direct power input to the module.

4.1.3 Addition of 5V-to-3.3V logic level conversion module

Since the Arduino Mega 2560 mainboard uses 5 V TTL serial logic levels, while the E27V-S module operates at 3–3.3 V TTL levels, we added this logic level conversion module (2 channels) to ensure proper voltage matching between the two devices. This prevents signal distortion or potential damage caused by voltage incompatibility, ensuring stable and reliable serial communication.

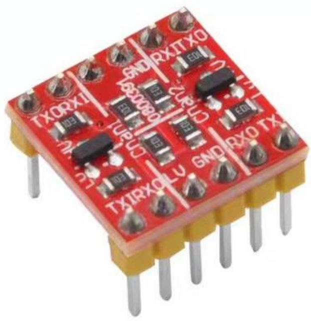


Figure 16 Appearance of Logic Level Conversion Module

This circuit schematic is as below, which primarily utilizes the conduction characteristics of N-channel MOSFETs (Q1, Q2) to achieve automatic level matching between different voltage logic systems.

The left side of the circuit is the low-voltage side (LV)—typically a 3.3 V system—while the right side is the high-voltage side (HV)—typically a 5 V system. When either side outputs a low-level signal, the MOSFET automatically conducts, creating a current path and enabling bidirectional signal transmission without requiring a direction control signal. Pull-up resistors (R1–R4) ensure that the signals remain high when idle, while R5–R8 serve as voltage dividers and buffers, improving conversion stability and noise immunity.

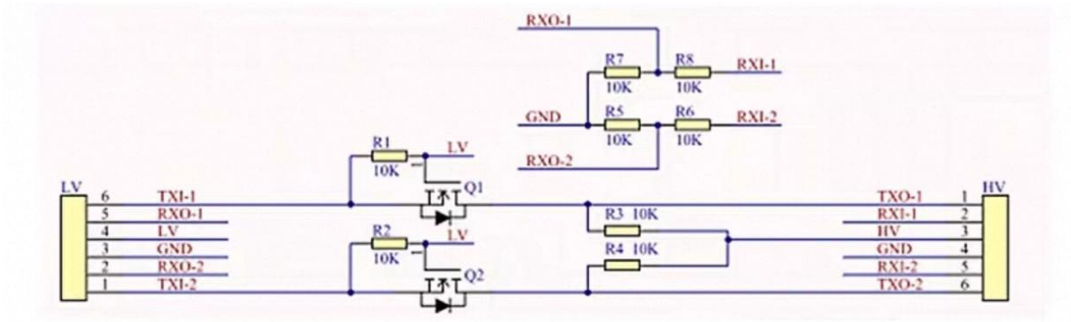


Figure 17 Circuit Schematic

4.2 Data Transmission Scheme

We select the Tastek DTU Cloud Operating Mode as our communication mode. In simple terms, it uses Tastek's 4G DTU module (TAS-LTE-4G) to upload data collected from field devices to the cloud, where the Tastek Cloud Platform enables remote access and control. We can set this module by AT commands on the configuration tool platform provided by Tastek.

❖ **Basic AT commands:**

Format:AT+DTUCLOUD=A,"B","C",D

Where:

A – DTU Cloud mode enable parameter (range 0–1):

0: Disable DTU Cloud; 1: Connect to Tastek DTU Cloud

B – Device ID (1–20 Bytes):

Automatically generated as an 8-digit ID when the device is created on the DTU Cloud platform.

C – Device Password (1–40 Bytes):
User-defined password entered during device creation on the DTU Cloud platform.

D – Channel selection (1–2):
Represents the two available socket channels.

Format: AT+DTUCLOUD?

The command is used to Query DTU Cloud Mode Configuration Details.

❖ **Detailed Process:**

The front-end sensors or devices, which can be our sensors and mainboard in our project, first transmit data via a serial interface to E27V-S module. The module then connects to the nearest communication base station using the 4G cellular network and transfers the data through the carrier network to the Tastek DTU Cloud Platform. On the cloud platform, the system performs device identification and authorization, data reception, processing, distribution, and storage.

On the user side, the cloud platform can be accessed either through a web interface or a virtual serial port, allowing users to monitor real-time data or send control commands. We choose virtual serial ports to realize communication. These commands are then relayed back through the network, base station, and DTU module to the field devices, achieving remote communication and control.

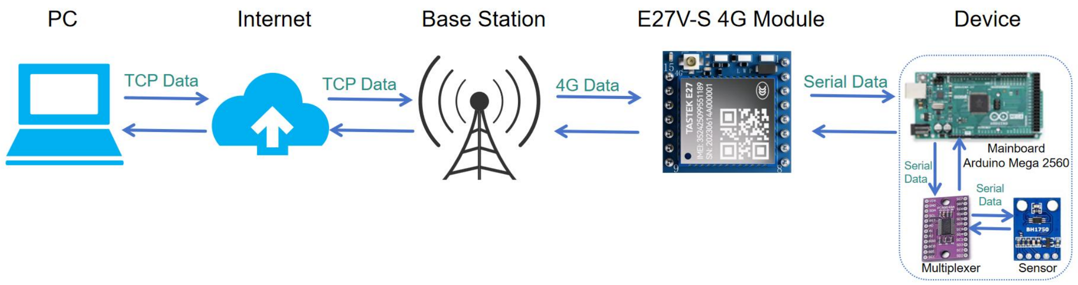


Figure 18 Data Flow Topology Diagram

❖ **Key Features:**

No dedicated server or public IP is required, as all communication is routed through the Tastek Cloud for seamless connectivity.

Device ID and device key authentication ensure secure connections, preventing unauthorized access.

With this setup, we can achieve stable and secure remote control between our device and PC end.

5. System Power and Signal Management

5.1 Wiring and Signal Transmission Requirements of Each Module

Since most modules require a 5V or 3.3V power supply, and considering the 220V AC power standard used in China, we selected a 220V-to-DC 5V voltage adapter. Its appearance and specifications are shown in the figure below.

Figure 19 Appearance and Mechanical Dimensions of Power Adapter

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)

As shown in the figure below, by connecting the live wire, neutral wire, and ground wire to the 220V AC power supply, the voltage adapter outputs two channels of DC 5V — labeled **+V** and **–V**. In addition, the converter is equipped with an indicator light that shows whether the converter is powered on.

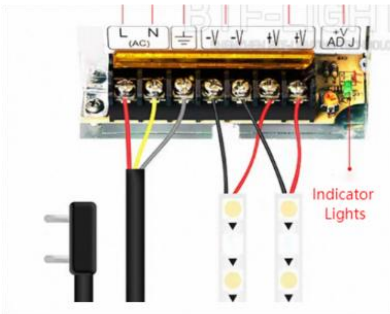


Figure 20 Wiring Diagram

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)

5.1.1 Arduino Mega 2560

The controller we selected is the Arduino Mega 2560 mainboard, and its pin diagram is shown below. In the actual wiring, a 5V DC power supply is provided to the board through the **VIN** and **GND** pins.

According to the program configuration, digital pin 6 (PWM) is used as the data output port to control the LED matrix data stream.

Digital pins 20 (SDA) and 21 (SCL) serve as the communication interfaces between the controller and the feedback module. These two pins are connected to four PCA9548A modules via a 1-to-4 Dupont cable, enabling I2C communication with a total of 25 GY-302 sensors.

Data transmission with the E27V-S 4G remote communication module is achieved through digital pin 0 (RX) and digital pin 1 (TX) on the Arduino Mega 2560.

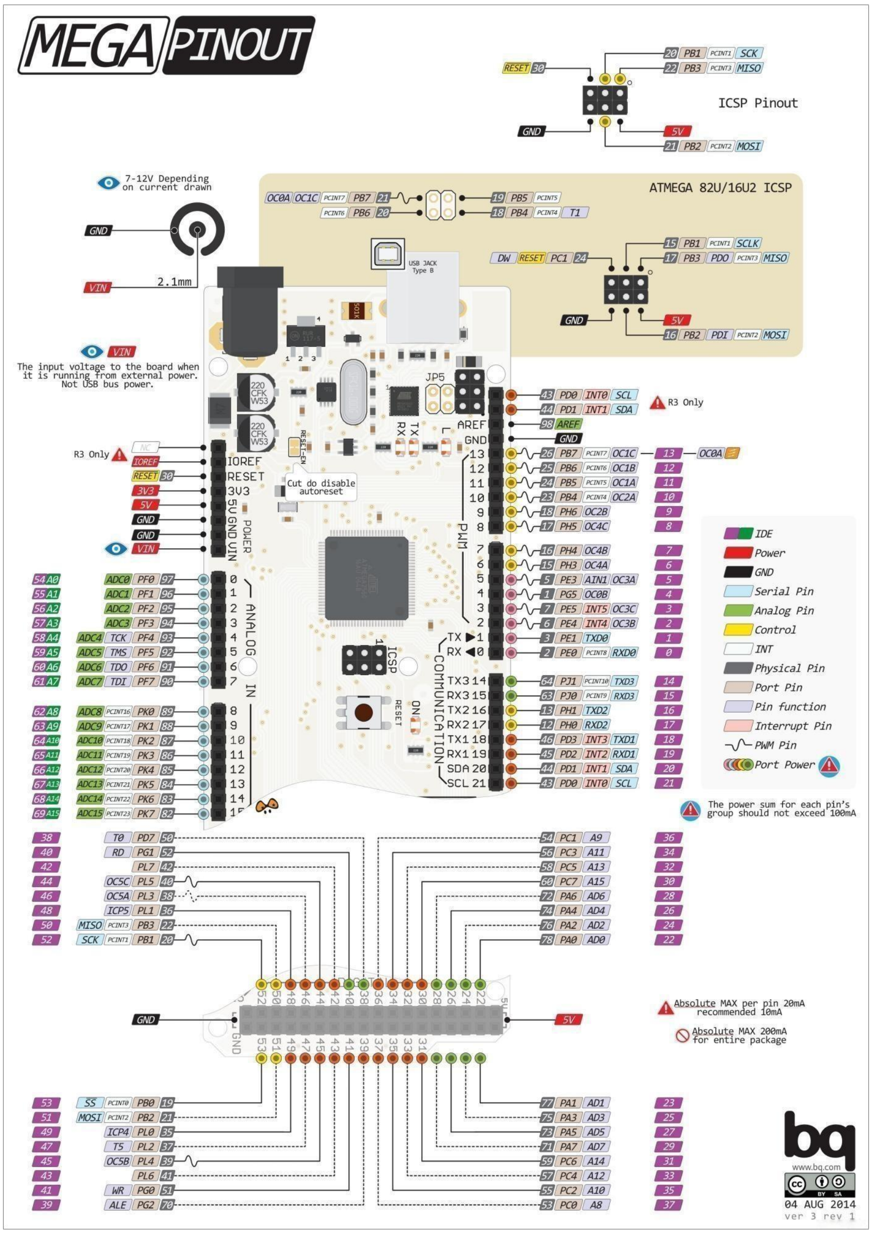


Figure 21 Pin Diagram of Arduino Mega 2560

The connection requirements for the main modules and components are as follows (since the dyeing module has not yet been finalized, only the wiring schemes for the other three modules are presented here):

5.1.2 WS2812 LED Light Array

The working voltage of the LED array is +3.7V to +5.5V. Each array is equipped with three pairs of **VCC** and **GND** input lines, any of which can be used to supply power to the array. These lines also serve as current-sharing and protective circuits.

Each LED array has one **DIN** line for data input and one **DOUT** line for data output. The **DOUT** line is only required when multiple LED arrays are connected in series. In our project, since only a single array is used, the **DOUT** line is not connected. The example wiring diagram is shown below.

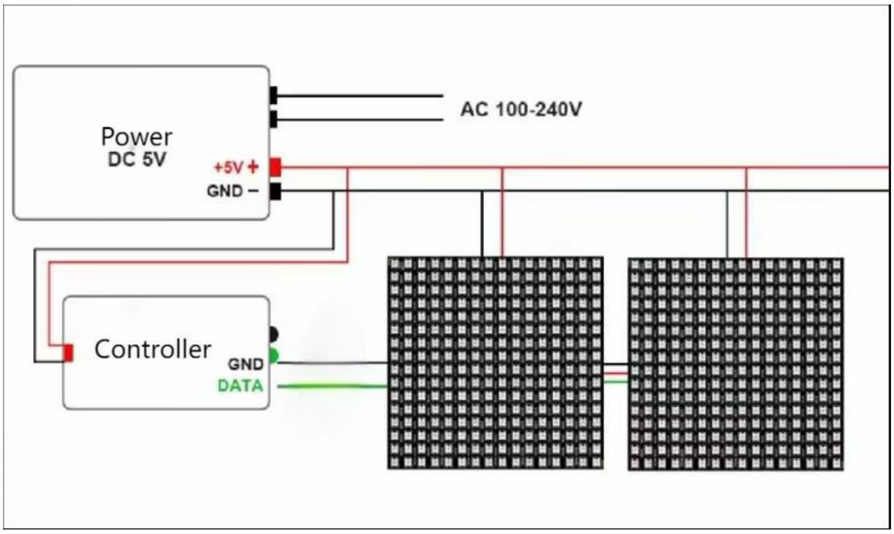


Figure 22 One Example Wiring Diagram of LED Light Array

5.1.3 GY-302 Light Intensity Sensor and PCA9548A

The GY-302 requires a 3.3V power supply, while the PCA9548A can operate with either 3.3V or 5V. To ensure communication stability and prevent voltage mismatch issues, we decided to power both the GY-302 and PCA9548A with a 3.3V supply.

5.1.4 E27V-S

The E27V-S operates reliably within a DC 5–16 V voltage range. Based on the actual device specifications, since most modules operate on 5V or 3.3V power, we chose to supply the E27V-S module with DC 5V to ensure stable operation and proper compatibility with the rest of the system.

5.2 Scheme for Power and Signal

As shown in the Figure 23, two independent 5V power supplies are obtained through the 220V-to-5V power adapter. Since the LED array must simultaneously drive 256 LED beads, which requires a relatively high current, one of the 5V outputs is dedicated exclusively to powering the LED array.

The second 5V output is connected to a breadboard, where it passes through a 5V-to-3.3V voltage converter to generate a 3.3V supply. This 5V and 3.3V power is then distributed via the breadboard to other modules in the system.

To ensure stable communication, the **GND** (ground) lines of all modules are shorted together, forming a common ground reference across the entire circuit.

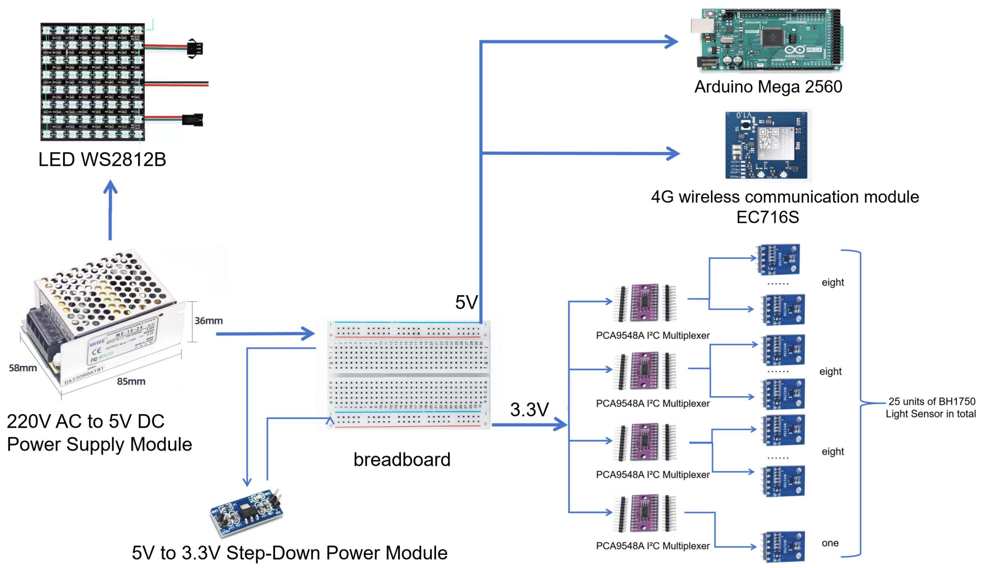


Figure 23 Power Scheme

As shown in Figure 24, the signal scheme is organized with the Arduino Mega 2560 serving as both the controller and the data center, facilitating data exchange among the three modules.

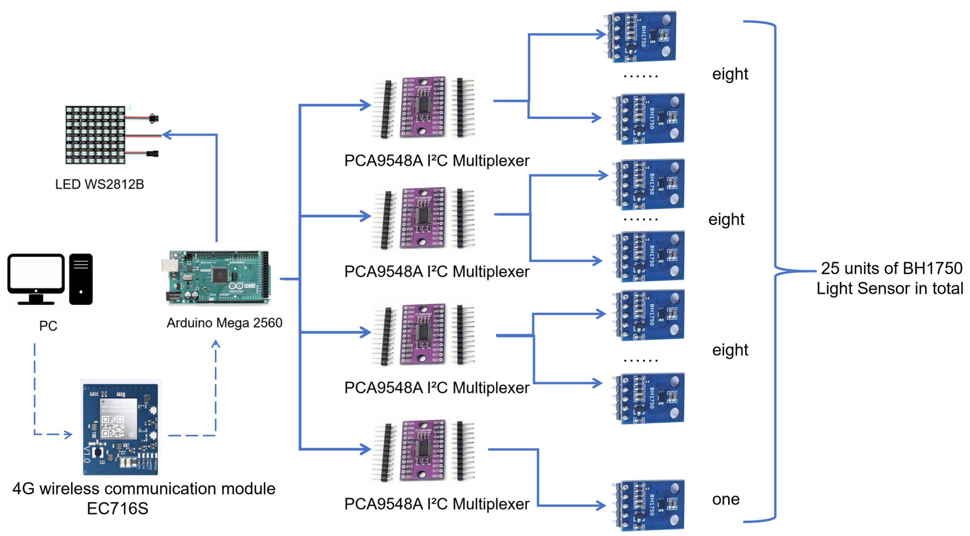


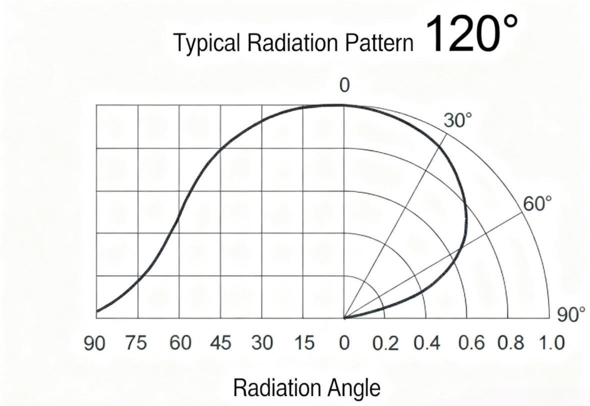
Figure 24 Signal Scheme

Appendix

Here are more details about our modules.

WS2812 LED Light Array

(Provided by SHENZHEN BTF-LIGHTING TECHNOLOGY CO., LTD)



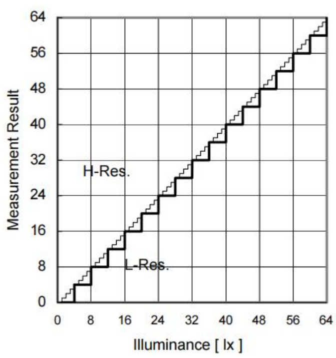
Typical Radiation Pattern

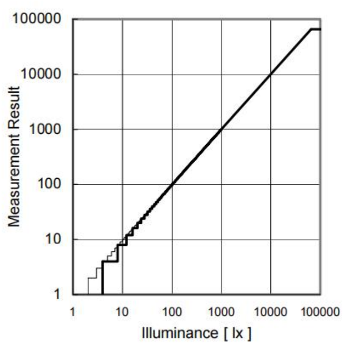
Parameter	Symbol	Range	Unit
Operaing Voltage	V_{DD}	+3.7 ~ +5.5	V
Logic Input Voltage	V_I	–0.5 ~ VDD+0.5	V
Operating Temperature	Topt	–40~+85	I ℃
Storage Temperature	Tstg	–40 +85	I ℃

Electrical Parameters

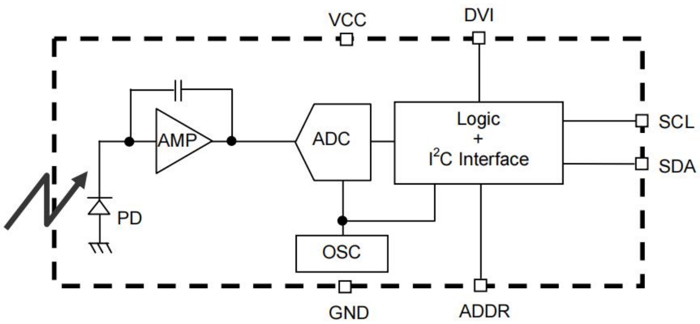
GY-302 Light Intensity Sensor (Chip of BH1750FVI)

(Provided by RHAM CO., LTD)





Illuminance Measurement Result



Block Diagram of BH1750FVI

Parameter	Symbol	Limits	Units
Supply Voltage	Vmax	4.5	V
Operating Temperature	Topr	-40～85	℃
Storage Temperature	Tstg	-40～100	℃
SDA Sink Current	Imax	7	mA
Power Dissipation	Pd	260※	mW

※ 70mm × 70mm × 1.6mm glass epoxy board. Derating in done at 3.47mW/℃ for operating above Ta=25℃.

Absolute Maximum Ratings

Parameter	Symbol	Min.	Typ.	Max.	Units
Vcc Voltage	Vcc	2.4	3.0	3.6	V
I²C Reference Voltage	VDVI	1.65	-	Vcc	V

Operating Conditions

PCB9548A

(Provided by TEXAS INSTRUMENTS)

over operating free-air temperature range (unless otherwise noted) (1)

		MIN	MAX	UNIT
V_{CC}	Supply voltage	–0.5	7	V
V_I	Input voltage(2)	–0.5	7	V
I_I	Input current	–20	20	mA
I_O	Output current	–25	25	mA
I_{CC}	Supply current	–100	100	mA
T_{stg}	Storage temperature	–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) The input negative-voltage and output voltage ratings may be exceeded if the input and output current ratings are observed.

Absolute Maximum Ratings

			MIN	MAX	UNIT
V_{CC}	Supply voltage		2.3	5.5	V
V_{IH}	High-level input voltage	SCL, SDA	0.7 × V_{CC}	6	V
		A2–A0, $\overline{RESET}$	0.7 × V_{CC}	V_{CC} + 0.5	
V_{IL}	Low-level input voltage	SCL, SDA	–0.5	0.3 × V_{CC}	V
		A2–A0, $\overline{RESET}$	–0.5	0.3 × V_{CC}	
T_A	Operating free-air temperature		–40	85	°C

(1) All unused control inputs of the device must be held at V_{CC} or GND to ensure proper device operation. See the *Implications of Slow or Floating CMOS Inputs* application report.

Recommended Operating Conditions

PIN NAME	PIN NO. DB, DW, DGV, PW	I/O	DESCRIPTION
A0	1	I	Address input 0. Connect directly to V_{CC} or ground
A1	2	I	Address input 1. Connect directly to V_{CC} or ground
$\overline{RESET}$	3	I	Active-low reset input. Connect to V_{CC} through a pull-up resistor, if not used
SD0	4	I/O	Serial data 0. Connect to V_{CC} through a pull-up resistor
SC0	5	I/O	Serial clock 0. Connect to V_{CC} through a pull-up resistor
SD1	6	I/O	Serial data 1. Connect to V_{CC} through a pull-up resistor
SC1	7	I/O	Serial clock 1. Connect to V_{CC} through a pull-up resistor
SD2	8	I/O	Serial data 2. Connect to V_{CC} through a pull-up resistor
SC2	9	I/O	Serial clock 2. Connect to V_{CC} through a pull-up resistor
SD3	10	I/O	Serial data 3. Connect to V_{CC} through a pull-up resistor
SC3	11	I/O	Serial clock 3. Connect to V_{CC} through a pull-up resistor
GND	12	—	Ground
SD4	13	I/O	Serial data 4. Connect to V_{CC} through a pull-up resistor
SC4	14	I/O	Serial clock 4. Connect to V_{CC} through a pull-up resistor
SD5	15	I/O	Serial data 5. Connect to V_{CC} through a pull-up resistor
SC5	16	I/O	Serial clock 5. Connect to V_{CC} through a pull-up resistor
SD6	17	I/O	Serial data 6. Connect to V_{CC} through a pull-up resistor
SC6	18	I/O	Serial clock 6. Connect to V_{CC} through a pull-up resistor
SD7	19	I/O	Serial data 7. Connect to V_{CC} through a pull-up resistor
SC7	20	I/O	Serial clock 7. Connect to V_{CC} through a pull-up resistor
A2	21	I	Address input 2. Connect directly to V_{CC} or ground
SCL	22	I/O	Serial clock bus. Connect to V_{CC} through a pull-up resistor
SDA	23	I/O	Serial data bus. Connect to V_{CC} through a pull-up resistor
V_{CC}	24	—	Supply voltage

Pin Functions

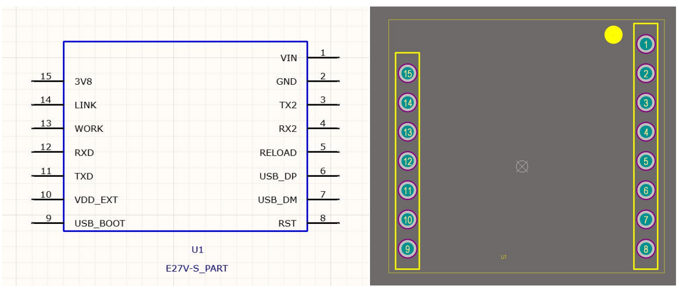
V_{CC} = 2.3 V to 3.6 V, over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS	V_{CC}	MIN	TYP(1)	MAX	UNIT
V_{PORR}	Power-on reset voltage, V_{CC} rising		No load, $V_I = V_{CC}$ or GND			1.2	1.5	V
V_{PORF}	Power-on reset voltage, V_{CC} falling(2)		No load, $V_I = V_{CC}$ or GND		0.8	1		V
$V_{o(sw)}$	Switch output voltage		$V_{i(sw)} = V_{CC}$, I_{SWout} = –100 µA	5 V		3.6		V
				4.5 V to 5.5 V	2.6		4.5	
				3.3 V		1.9		
				3 V to 3.6 V	1.6		2.8	
				2.5 V		1.5		
				2.3 V to 2.7 V	1.1		2	
I_{OL}	SDA		V_{OL} = 0.4 V	2.3 V to 5.5 V	3	6		mA
			V_{OL} = 0.6 V		6	9		
I_I	SCL, SDA		$V_I = V_{CC}$ or GND	2.3 V to 5.5 V	–1		1	µA
	SC7–SC0, SD7–SD0				–1		1	
	A2–A0				–1		1	
	$\overline{RESET}$				–1		1	
I_{CC}	Operating mode	f_{SCL} = 400 kHz	$V_I = V_{CC}$ or GND, I_O = 0	5.5 V		50	80	µA
				3.6 V		20	35	
				2.7 V		11	20	
		f_{SCL} = 100 kHz	$V_I = V_{CC}$ or GND, I_O = 0	5.5 V		9	30	
				3.6 V		6	15	
				2.7 V		4	8	
	Standby mode	Low inputs	V_I = GND, I_O = 0	5.5 V		0.2	2	
				3.6 V		0.1	2	
				2.7 V		0.1	1	
		High inputs	$V_I = V_{CC}$, I_O = 0	5.5 V		0.2	2	
				3.6 V		0.1	2	
				2.7 V		0.1	1	
ΔI_{CC}	Supply-current change	SCL, SDA	SCL or SDA input at 0.6 V, Other inputs at V_{CC} or GND	2.3 V to 5.5 V		3	20	µA
			SCL or SDA input at V_{CC} – 0.6 V, Other inputs at V_{CC} or GND			3	20	
C_i	A2–A0		$V_I = V_{CC}$ or GND	2.3 V to 5.5 V		4	5	pF
	$\overline{RESET}$					4	5	
	SCL		$V_I = V_{CC}$ or GND, Switch OFF			20	28	
$C_{io(off)}$ (3)	SDA		$V_I = V_{CC}$ or GND, Switch OFF	2.3 V to 5.5 V		20	28	pF
	SC7–SC0, SD7–SD0					5.5	7.5	
R_{ON}	Switch-on resistance		V_O = 0.4 V, I_O = 15 mA	4.5 V to 5.5 V	4	10	20	Ω
				3 V to 3.6 V	5	12	30	
			V_O = 0.4 V, I_O = 10 mA	2.3 V to 2.7 V	7	15	45	

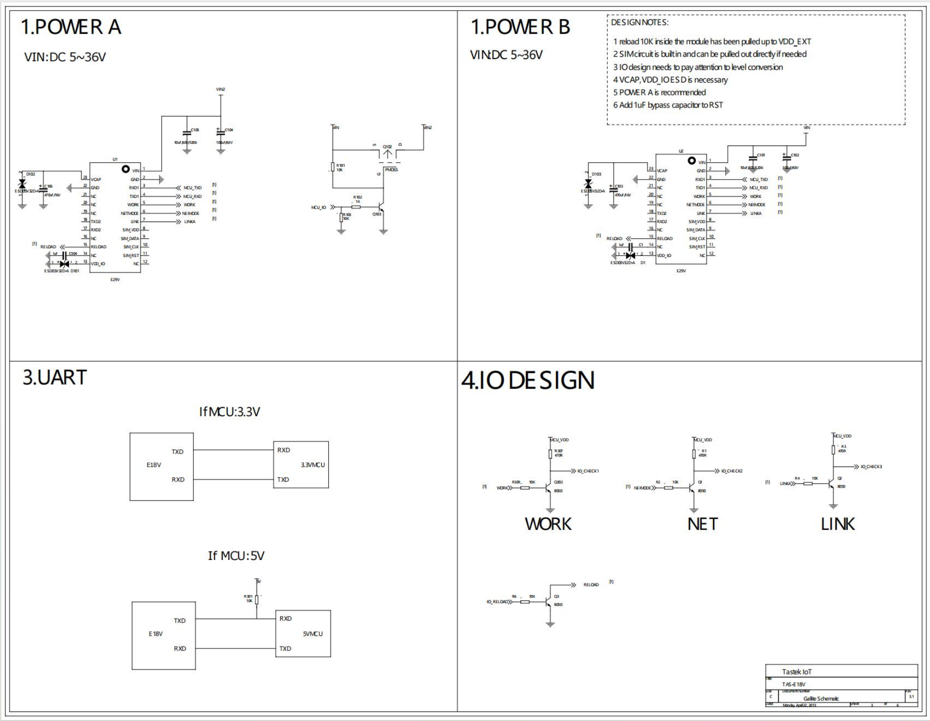
(1) All typical values are at nominal supply voltage (2.5-, 3.3-, or 5-V V_{CC}), T_A = 25°C.
(2) The power-on reset circuit resets the I²C bus logic with $V_{CC} < V_{PORF}$.
(3) $C_{io(ON)}$ depends on internal capacitance and external capacitance added to the SCn lines when channels(s) are ON.

Electrical Characteristics

E27V-S (Provided by Tastek IoT)



Pin Design and Packaging (In Altium Designer)



Hardware Reference Design

Category	Item	Content
-	Product Name	E27V-S
CPU	CPU Core	Cortex-M3
	Main Frequency	204M
Wireless Parameters	Standard	CAT.1
	Protocol Version	3GPP R14
	Frequency Bands	LB: 617 MHz ~960 MHz MB: 1710 MHz ~2200 MHz HB: 2300 MHz ~2690 MHz
	Transmit Power	23dBm±2.7dB (Max)

Basic Parameters-1

	Antenna Interface	IPEX-01*1
Hardware Parameters	Operating Voltage	DC 3.8V/DC 5~16V
	Operating Temperature	-30~75℃
	Operating Current	35mA@12V
	UART	2 (TTL 3.0V~3.3V Level)
	Baud Rate	1200-115200
	Data Bits	7/8
	Parity Bit	None Odd Even
	Stop Bits	1, 2
	Dimension	2321.512mm (LWH, including Pins)

Basic Parameters-2

Item	Content
Virtual Serial Port Tastek-VCOM	WIN7/WIN10
Configuration Method	Serial Port Configuration, Host Computer Setting, Network Configuration
Network Protocols	TCP, UDP, DNS, MQTT, HTTP, WebSocket, NTP
Number of Network Channels	2
Network Cache	2 Channels Share 50K/50 Entries
SMS	Support SMS Configuration
Service Heartbeat	Custom Heartbeat
Service Registration Packet	Support Custom Registration Packet/IMEI/ICCID

Basic Parameters-3