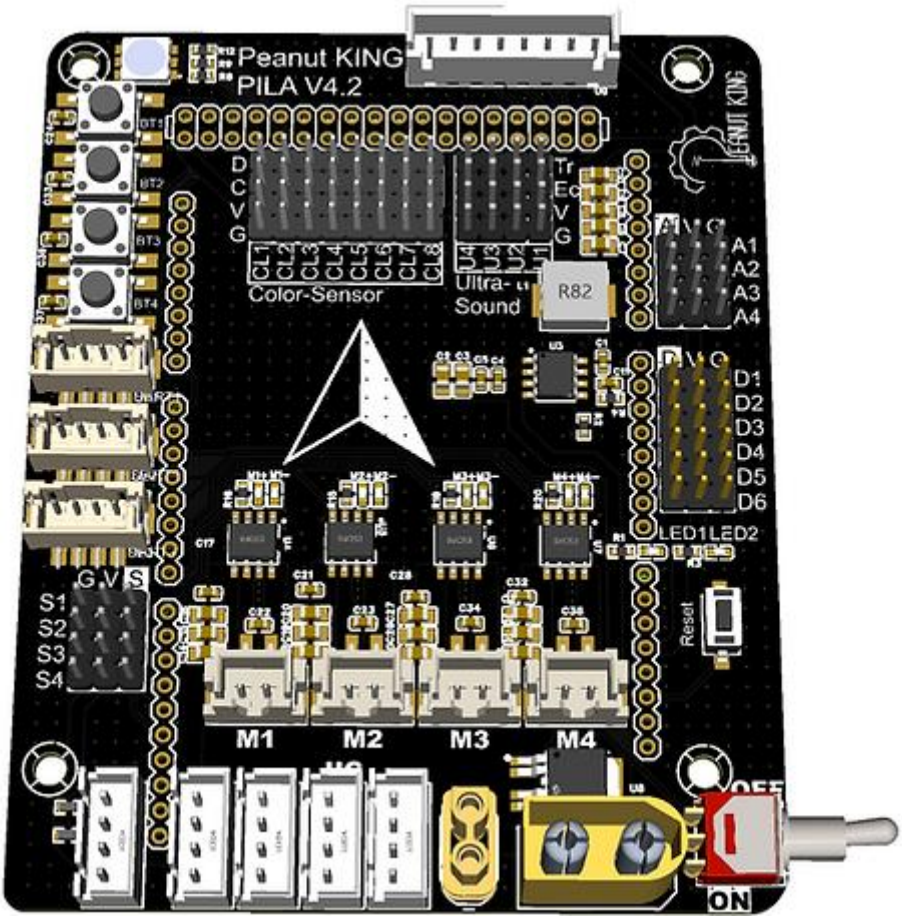
PILA V4



Introduction

The PILA V4 is an extension module designed for use with the Arduino Mega to significantly enhance usability and extensibility. This shield helps users develop different projects easily and efficiently. It features 4 AT8236 motor drivers, offering 4 motor ports with each is capable of up to 6 A peak current per way.

The shield supports 5V VCC Input/Output and utilizes an XT60 Port for power supply. Additionally, it features 8-Channel software I^2C, allowing the PILA V4 Shield to provide a total of 13 I^2C interfaces (8 software + 5 hardware).

Features

On Board RGB LED	: 1
Serial interface(UART)	: 3
Motor interface	: 4
Servo interface	: 4
Hardware I^2C interface	: 5
OLED interface	: 1
Analog Input Port	: 10
Ultrasound Port	: 4
Color Sensor Port (Software I^2C)	: 8
Button	: 4
Reset Button	: 1

Specification

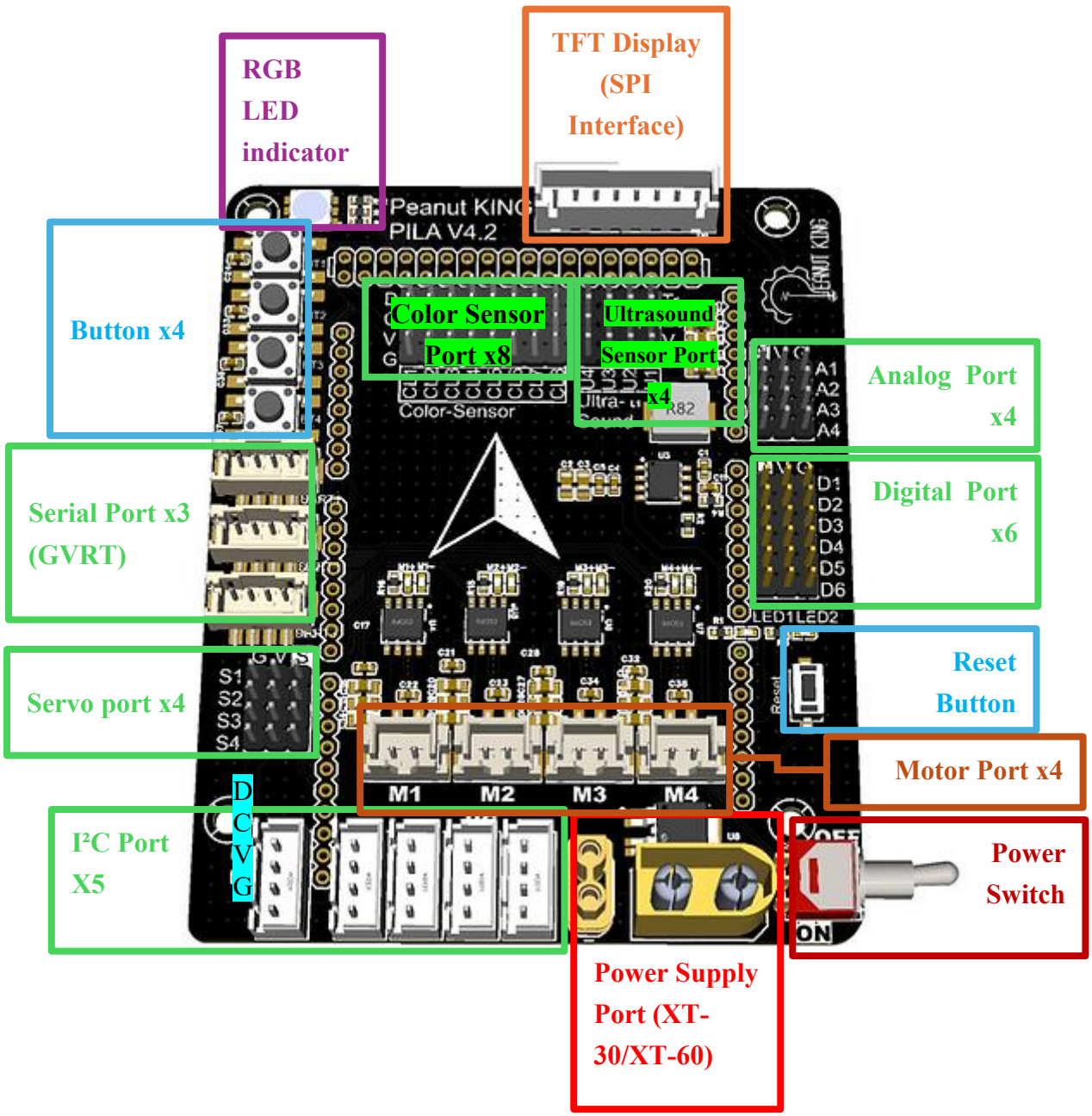
Power supply voltage	: 4.5 ~ 12V DC
Logic Control Voltage	: 5V (From Arduino)
Operating temperature	: −40 ~ 85°C
Operating Humidity	: < 90%
Weight	: 60 ± 1 g
Size	: 75 x 110 x 50 mm

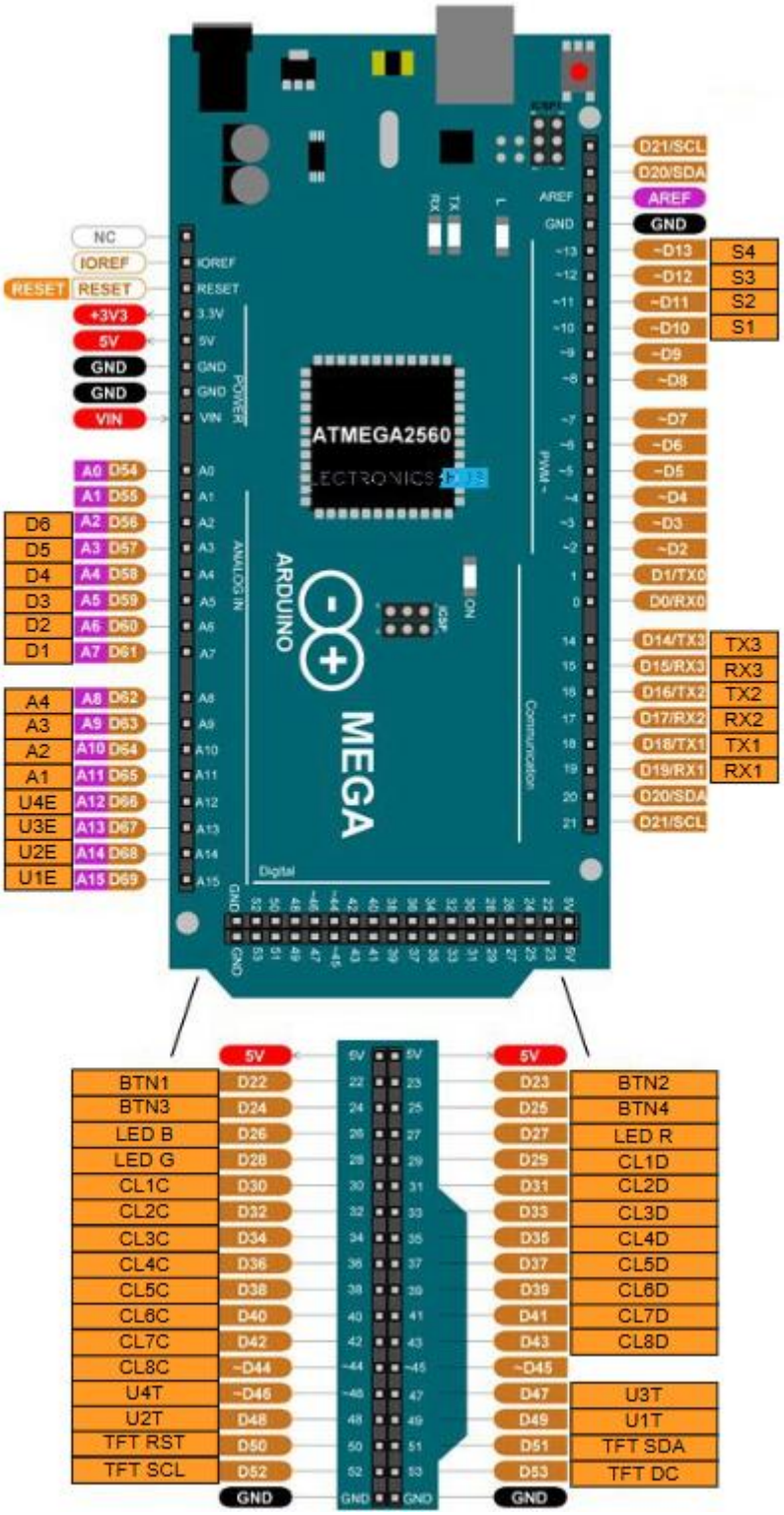
2-way motor drive

Up to 6 A peak current each way

Support PWM speed control (490.2Hz)

PILA V4 Shield

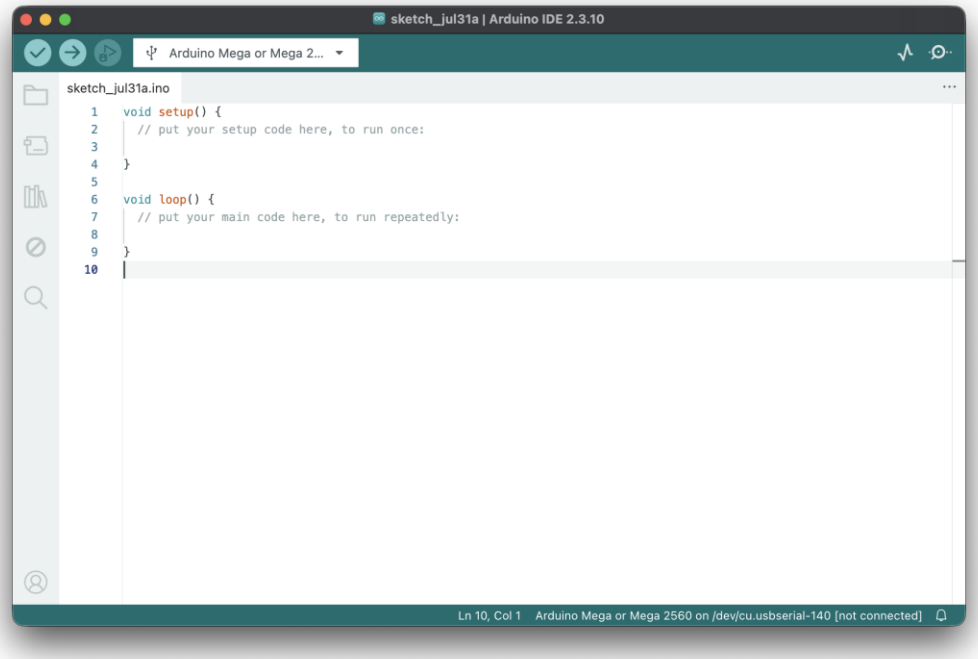




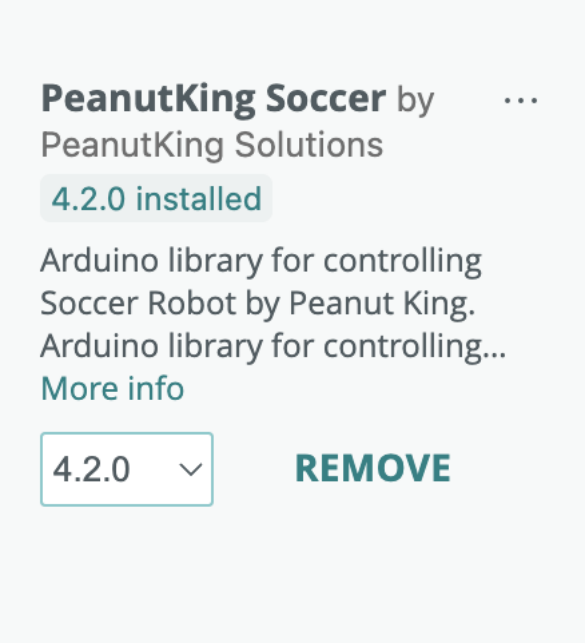
Application and Setup

Plug the Arduino Mega into the Peanut KING PILA V4 according to the pin markings on the shield. You must ensure that the pins on the shield exactly match the pins on the Arduino Mega.

Open a new sketch in the Arduino IDE



Install the PeanutKing Soccer library Version 4.2.0.

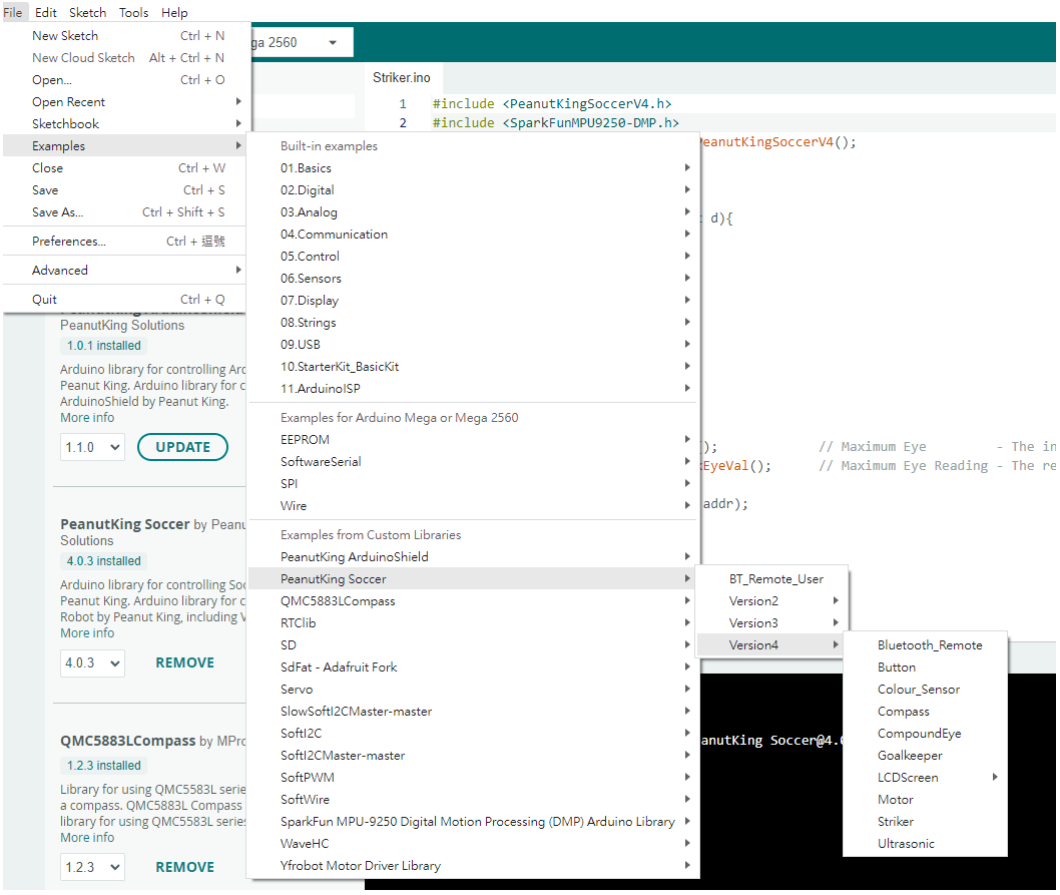


-

Sample Code Access

Once the PILA V4 is successfully connected to the Arduino Mega and the required library is installed, you can access several pre-configured examples.

To find these examples in the Arduino IDE, navigate to: **Files >> Examples >> PeanutKing Soccer >> Version4**.



Change Log from V3

Added:

- **compoundMaxEye()**: This function returns the maximum eye index, ranging from 0 to 11.
- **compoundMaxEyeVal()**: This function returns the maximum eye reading, ranging from 0 to 255.
- **getColorSensor(CL_no)**: This function reads the result of the color sensor. The parameter CL_no refers to the CL1~CL8 on the board. It uses the following enum to represent colors:

```
typedef enum {
  BLACK,
  WHITE,
  GREY,
  RED,
  GREEN,
  BLUE,
  YELLOW,
  CYAN
}color_sensor_color;
```

- **HSL_Struct getColorSensorHSL(CL_no)**: This function reads the HSL data of the color sensor. The parameter CL_no refers to the CL1~CL8 on the board.
- **RGB_Struct getColorSensorRGB(CL_no)**: This function reads the RGB data of the color sensor. The parameter CL_no refers to the CL1~CL8 on the board.
- Easier access to use analogRead(), analogWrite(), digitalRead(), digitalWrite() by the following new enum:

```
typedef enum{
  S1_P = 10,
  S2_P,
  S3_P,
  S4_P,
}S_PIN;
```

```
typedef enum{
  D6_P = 56,
  D5_P,
  D4_P,
  D3_P,
  D2_P,
  D1_P,
}D_PIN;
```

```
typedef enum{
  A4_P = 62,
  A3_P,
  A2_P,
  A1_P,
}A_PIN;
```

Changed:

- Modified **compoundEyeRead()** function to return a uint8_t* pointer to array instead of a single uint8_t.
- Usage example: uint8_t* eye = **compoundEyeRead()**; where eye[eye_no], with eye_no ranging from 0 to 11.

Change the lcd to big tft screen:

```
void
  printSpace(uint32_t, uint8_t digits = 3),
  setScreen(uint8_t, uint8_t, char[] ),
  setScreen(uint8_t, uint8_t, int16_t, uint8_t digits = 3),
  lcdSetup(void),
  lcdClear(void),
  setCursor(uint8_t col, uint8_t row);
```

Software I²C

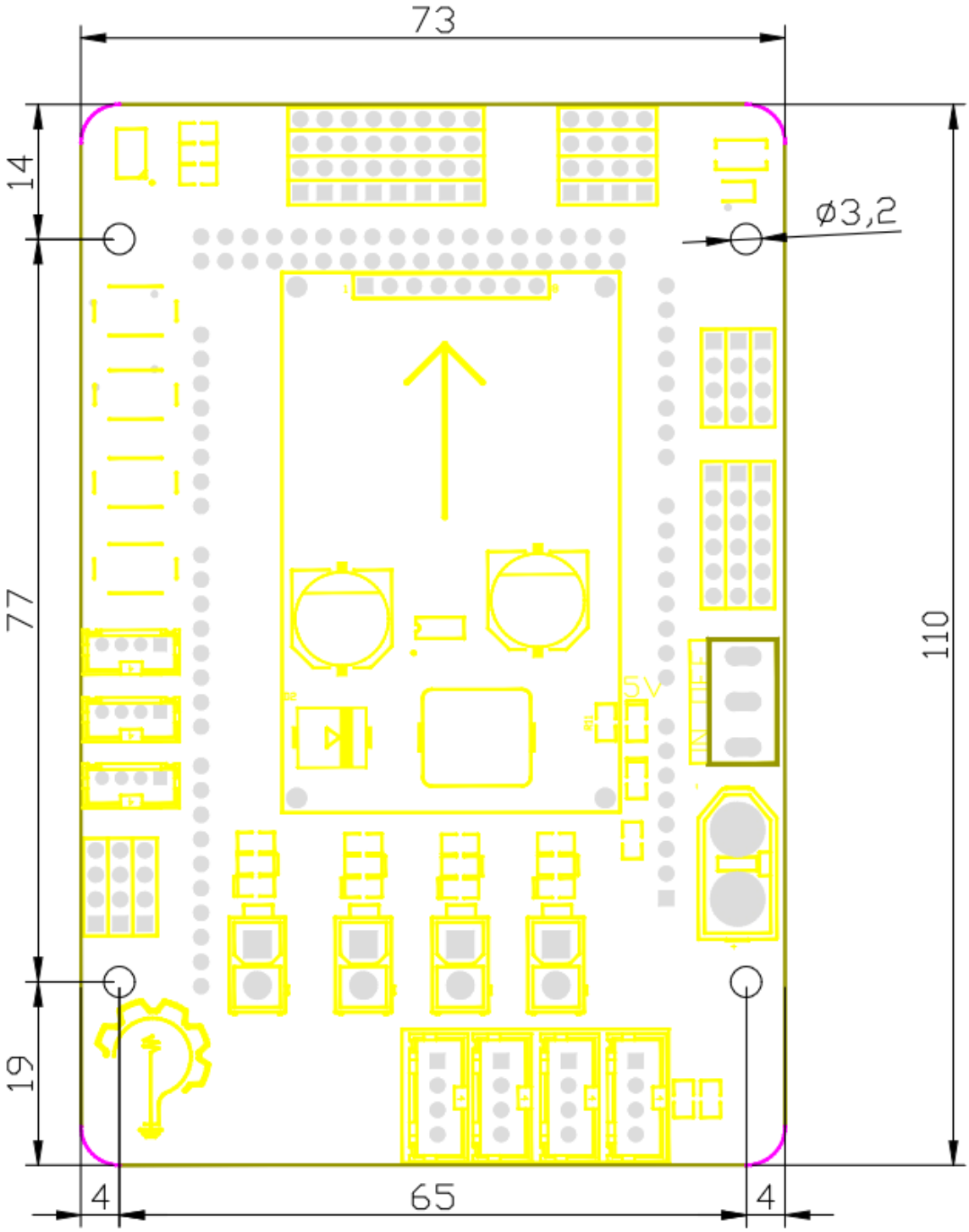
The Color Sensor port uses software I²C. Therefore, 8 sensors with same address could be communicated through I²C. If there are more than 4 sensors to connect with the hardware I²C, you can also try to use robot.swiic[i].i2cxxx(); to communicate with the sensor using swiic.

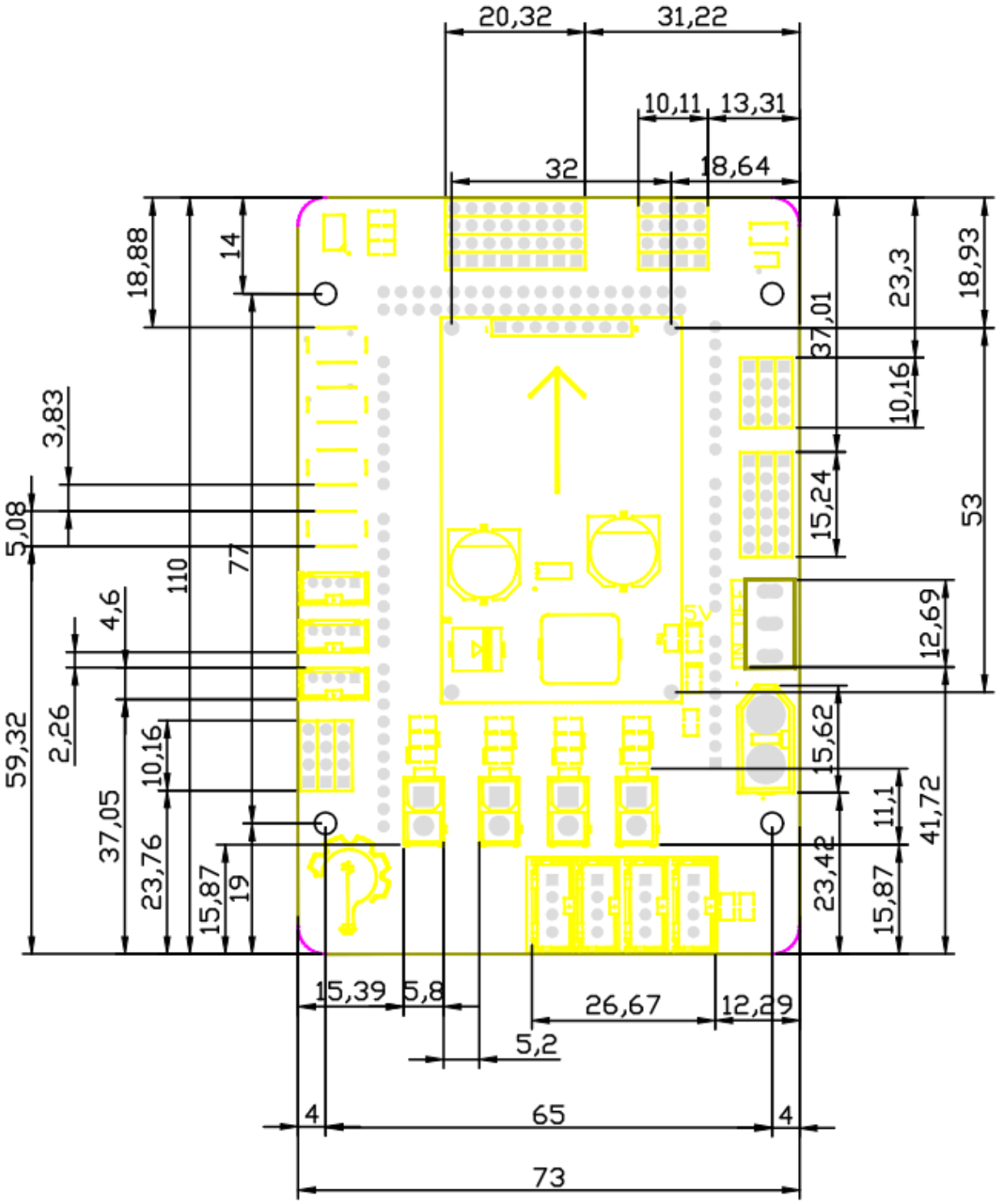
```
robot.swiic[0].i2c_start(I2C_7BITADDR<<1){
  robot.swiic[0].i2c_write(I2C_REGISTER);
  robot.swiic[0].i2c_rep_start((I2C_7BITADDR<<1)|I2C_READ);
  byte val = robot.swiic[0].i2c_read(true);
  robot.swiic[0].i2c_stop();
}
```

Wire.h

This library is not compatible with "Wire.h". It is specifically designed to provide full support exclusively for our company's products. To use certain online open-source libraries, we strongly recommend reimplementing the I²C function for communication with our products using "Wire.h", as this approach should be more straightforward than modifying alternative libraries. For I²C addresses and register details of our sensors, please refer to our other datasheets.

Appendix Dimension





Revision Information

Change from 1.0 (august ,2024) to current version	Pages
Initial release	12

Important Information and Disclaimer

The information in this statement reflects Peanut King Solution Limited's knowledge and belief as of the date it is provided. Peanut King Solution Limited relies on information from third parties and does not guarantee its accuracy. Efforts are ongoing to better integrate third-party information. Peanut King Solution Limited has taken and continues to take reasonable steps to provide accurate and representative information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. Certain information is considered proprietary by Peanut King Solution Limited and its suppliers, so CAS numbers and other limited information may not be available for release.

NOTICE

Peanut King Solution Limited reserves the right to make changes to the products in this document to improve performance or for any other reason, or to discontinue them without notice. Customers should contact Peanut King Solution Limited for the latest product information before placing orders or designing Peanut King Solution Limited products into systems. Peanut King Solution Limited assumes no responsibility for the use of any products or circuits described in this document or customer product design, does not convey any license under any patent or other right, and does not guarantee that the circuits are free of patent infringement. Peanut King Solution Limited makes no claim regarding the suitability of its products for any particular purpose and assumes no liability arising from the use of any product or circuit, specifically disclaiming any and all liability, including consequential or incidental damages.

PEANUT KING SOLUTION LIMITED PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY OR DEATH. THE USE OF PEANUT KING SOLUTION LIMITED PRODUCTS IN LIFE SUPPORT SYSTEMS IS NOT AUTHORIZED.