WRO Future Engineer 2025

Documentation

By

Chaiwat Chinsupawat

Pongpapat Putongkam

Peradon Nimsongprasert

Introduction

We are a group of students from Yothinburana School who share a strong passion for programming and robotics. All three of us are active members of the school's Robotics Club, which has around 30 members. We each joined the club in Grade 7 (Mathayom 1), driven by a love for technology and building things from scratch.

Our team is called "F1", inspired by our enthusiasm for Formula 1 racing. Beyond just being fans, we dream of one day designing and building our own self-driving vehicle — a goal that keeps us motivated and united as a team. Our goal is to innovate, build, and one day revolutionize autonomous vehicles. Every challenge we face is an opportunity to grow — not only as engineers, but also as a team united by the same vision.

Robot Aim and Objectives

Aim

The aim of our robot project is to design, build, and program an innovative robotic system that can address real-world challenges while showcasing creativity, teamwork, and engineering skills. This project allows us to bridge theoretical STEM knowledge with practical applications, encouraging us to think critically and develop unique solutions.

Objectives

1. **Innovation and Creativity**
   - Design and build robots capable of performing complex tasks.
   - Promote creative thinking and the development of unique solutions to given challenges.
2. **Engineering Skills**
   - Strengthen our knowledge of mechanical design, electronics, and programming.

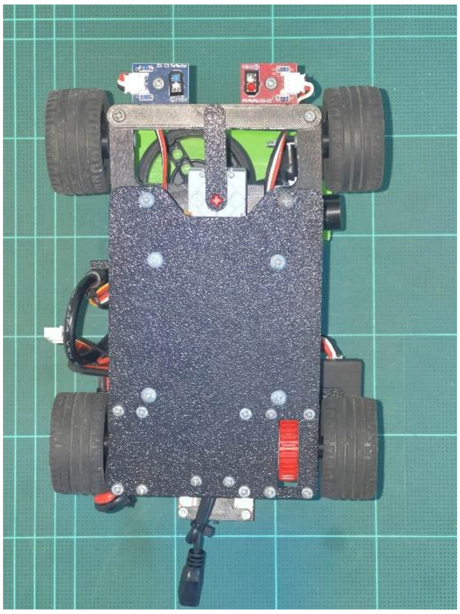
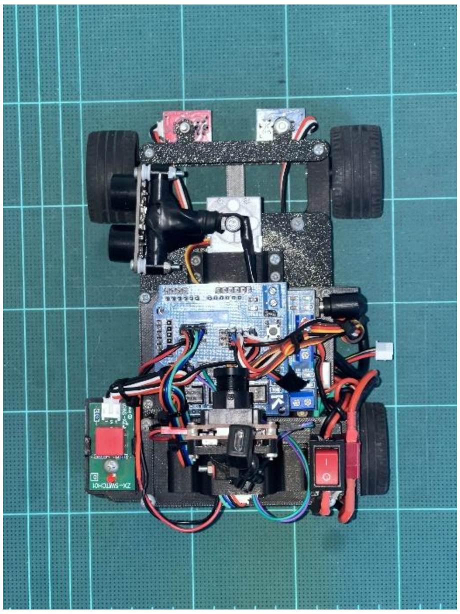
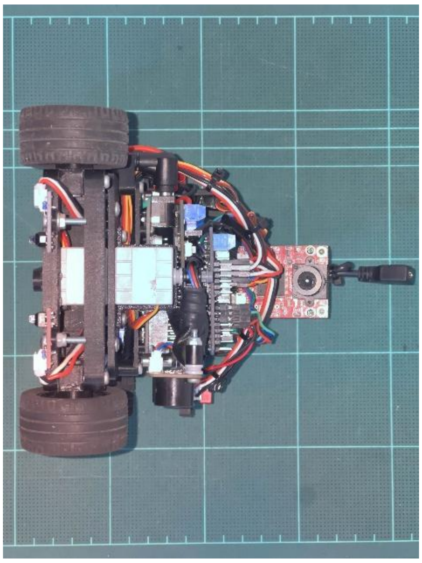
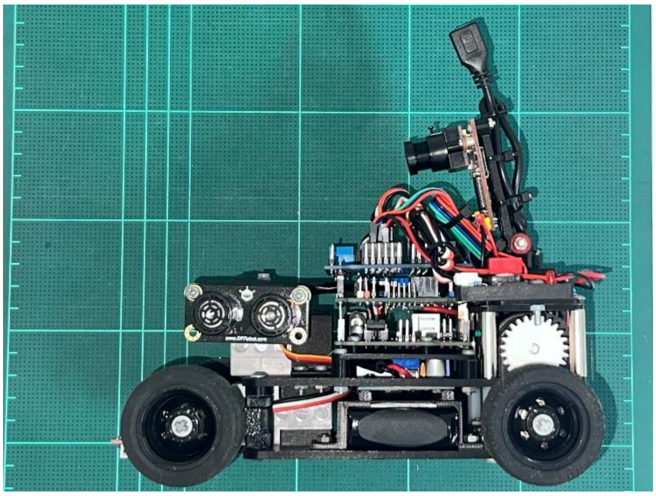
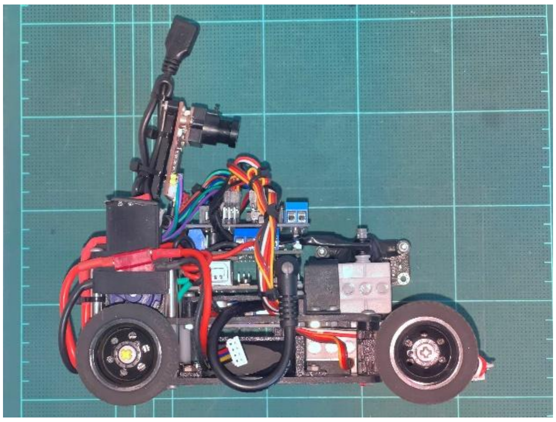
- Integrate hardware and software into a functional robotic system.

3. **Problem-Solving**
   - Apply analytical thinking and strategic planning to solve real-world challenges.
   - Develop troubleshooting and optimization skills for improving robot performance.
4. **Collaboration and Teamwork**
   - Work effectively as a team by fostering communication and cooperation.
   - Build strong collaboration skills to achieve shared goals together.
5. **Application of STEM Knowledge**
   - Apply science, technology, engineering, and mathematics concepts to practical robotics scenarios.
   - Integrate multiple STEM disciplines to develop comprehensive robotic solutions.
6. **Presentation and Documentation**

- Enhance our ability to present projects clearly and confidently.
- Emphasize the importance of thorough documentation, including the design process, programming, and testing

Team Photo

Robot Picture

Robot Design

The Robot Designing a robot for this competition is sometimes hard. We need to design it to be compact, lightweight, durable, and have all the functions we need. The robot must be four-wheeled with a steering function on either front or back wheel, which is why we need to be careful to pick the components. After we select all the components we need, then we think about how to put them all together. We came up with using Lego Technic parts. We spent a lot of time designing it to be within the size regulation. Designing a robot using Lego parts is also not easy since we need to think about everything. Every part we put in includes weight, which makes the robot slower. We failed a few times before coming up with the design we use now. After finishing designing, we combined it with other components to see the result. he robot's color also shouldn't be green and red since the camera can confuse it with the obstacles. For the design, we placed the gyro around the middle-top of the robot to get the most accurate output, while the camera was positioned at the front of the gyro to detect objects quickly. Our robot has two servos, one on the bottom front and one on the top front. The bottom front servo is used to steer the wheel. Then, we added a top servo that rotates our ultrasonic. We use an ultrasonic to keep our robot out of the wall. Finally, we place our motor at the bottom back of the robot, and to ensure that it drives both wheels at the back, we use a gear differential from Lego.

Design Software

At the beginning of this year's design process, we experimented with five different design tools to explore their capabilities and limitations. While Blender offered a user-friendly and intuitive interface, we realized that it lacked the precision required for engineering-grade robotics. For the first time, we made the shift to a professional-grade CAD tool — Autodesk Fusion 360.

This decision was driven by the increasing complexity of our robot's hardware. As we upgraded our electronic components, our mechanical structures also needed to achieve tighter tolerances and higher stability. Fusion 360 enables us to design with millimeter-level accuracy, which is critical for ensuring that sensors, motors, and moving parts align perfectly. Furthermore, its integrated simulation tools allow us to test stress points, weight distribution, and kinematics virtually, reducing trial-and-error during physical assembly.

By combining CAD precision with our team's 3D design expertise, we have created a robot that not only looks refined in digital models but also translates seamlessly into a high-performance physical build.

Design Hardware

Bambu Lab X1C 3D Printer

We chose the Bambu Lab X1C 3D Printer because it is widely recognized and popular among students and beginners due to its ease of use. Unlike many professional 3D printers, it is not complex, making it ideal for learning and rapid prototyping. Despite its simplicity, the print quality is professional-grade, delivering highly detailed and precise results. Additionally, the printer is affordable relative to the performance it provides, offering excellent value for money compared to other printers on the market. This combination of usability, quality, and cost-effectiveness makes it a perfect choice for both educational and practical applications.

🖨️ Product Features of Bambu Lab X1C

- 🎨 Multi-Color & Multi-Material Capability – Print with different colors and materials in a single job.
- ✅ Proven Reliability – Trusted by professionals worldwide.
- 📏 High-Quality Printing – Achieves 7 μm Lidar resolution for exceptional detail.
- ⚡ High-Speed CoreXY – Up to 20,000 mm/s^2 acceleration for rapid prints.
- 🛠️ Dual Auto Bed Leveling – Ensures perfect first layers every time.
- 🔄 AMS 2 Pro Compatibility – Works seamlessly with X1/P1 series printers for multi-material printing.

X1C Tech Specs

Feature	**Specification**
Build Volume (W×D×H)	**256 × 256 × 256 mm^3***
Nozzle	**0.4 mm Hardened Steel Included**
Hotend	**All-Metal**
Max Hot End Temperature	**300 °C**
Filament Diameter	**1.75 mm**

Feature	Specification
Supported Filament	PLA, PETG, TPU, ABS, ASA, PVA, PET Ideal for PA, PC, Carbon/Glass Fiber Reinforced Polymer
Build Plate Surface	Bambu Textured PEI Plate (Pre-installed, Random, Both compatible with Micro Lidar)
Max Build Plate Temperature	110 °C@220V, 120 °C@110V
Max Speed of Tool Head	500 mm/s
Max Acceleration of Tool Head	20 m/s²
Physical Dimensions	X1C: 389 * 389 * 457 mm³ Package size: 480 * 480 * 535 mm³ Net weight: 14.13 kg, Gross weight: 18 kg X1C Combo: Package size: 480 * 480 * 590 mm³ Gross weight (AMS included): 22.3 kg
Electrical Requirements	100–240 VAC, 50/60 Hz, 1000W@220V, 350W@110V

Mobility Management

1. Chihai Motor 25-370K

For our project, we selected the SPG20HP-20K, also widely known as the GM25 motor. This is a compact DC geared motor paired with a 20:1 gearbox, capable of delivering up to 225 mN·m of torque with a no-load speed of 980 RPM. Its combination of torque and speed makes it highly suitable for small-sized mobile robots, sumo robots, and various automation projects. The motor's standard size and the availability of different gear ratios allow it to be easily swapped without requiring significant modifications to the robot's mechanical design.

The rated operating voltage for the SPG20HP-20K is 12V. Operating it at a lower voltage can lead to insufficient torque, which may prevent the robot from effectively driving wheels or mechanisms, especially under load. Conversely, running it at a higher voltage can increase torque and speed, but may also accelerate wear and reduce the motor's overall lifespan. Careful consideration of voltage is therefore critical to balance performance and durability.

Initially, our team experimented with the LEGO™ Power Functions Large Motor as the primary drive motor. While this motor worked well for basic movement, we identified a critical limitation during testing: when reducing speed, the motor failed to deliver sufficient torque to reliably move the robot. This was particularly problematic for tasks that required precise control or gradual acceleration, which are essential in competitive challenges such as the WRO Future Engineers 2025.

To overcome this limitation, we transitioned to the Chihai Motor 25-370K. This motor provides higher torque and consistent performance at variable speeds, ensuring stable movement even under heavier loads. Its reliability allowed us to optimize mobility management—our robot can now navigate the course smoothly, maintain accurate positioning, and execute complex maneuvers without risk of stalling or slipping.

Overall, the selection of the Chihai Motor 25-370K reflects a careful balance between torque, speed, reliability, and compatibility with our robot design. By addressing the limitations of the LEGO motor and leveraging the strengths of the Chihai motor, we were able to enhance our robot's performance, stability, and competitiveness, which are crucial for success in the WRO Future Engineers challenge.

Feature / Motor	LEGO™ PF Large Motor	Chihai Motor
Type	DC Motor	DC Geared Motor
Rated Voltage	9V	12V
No-Load Speed	~170 RPM	980 RPM
Max Torque	Moderate	225 mN·m
Gear Ratio	Fixed	20:1
Size / Compatibility	LEGO standard	Compact, standard size
Performance at Low Speed	Poor torque, may stall	Moderate; better than LEGO
Reliability under Load	Limited	Moderate
Suitability for Obstacle Challenges	Limited	Moderate
Ease of Replacement	Easy in LEGO system	Easy with standard mounting

Feature / Motor	LEGO™ PF Large Motor	Chihai Motor
Notes / Observations	Works for basic movement; stalling on slow speed & sharp turns	Adequate for light robots, limited torque at slow speed

Motor comparison table (LEGO PF vs Chihai):

Specification:

Specification	Details
Torque	0.3 – 8.0 kg·cm
Model Number	CHR-GM25-370K
Continuous Current (A)	320 – 600 mA
Construction	Permanent Magnet
Type	Geared Motor
Commutation	Brushed
Brand Name	CHIHAI MOTOR
Efficiency	IE2
Place of Origin	China (CN)
Certification	CE
Application	Household Appliances

Product Description:

Product Type: CHR-GM25-370K dc gear motor

Output shaft length:12mm D

Shaft diameter:4mm

Voltage range:DC1.5- 12.0V

Weight:About 90 g

1. Pure copper winding:The rotor part is made of pure copper, with small resistance and strong electrical conductivity, which

increases the motor torque.

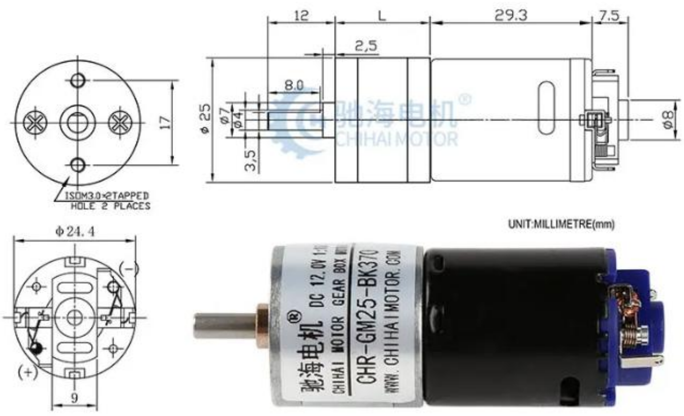
2. D type shaft: more convenient to install,not easy to skid,save installation time.

3. Extended carbon brush:tool extended carbon brush,replaceable

PRODUCT INFORMATION

Brand	CHIHAI MOTOR
Model	CHR-GM25-370K Gear Motors
Output Shaft Length	12mm D
Shaft Diameter	4mm
Voltage Range	DC1.5- 12.0V
Weight	About 90 g

Size



Detail

Pure Copper Winding

D Type Shaft

Extended Carbon Brush

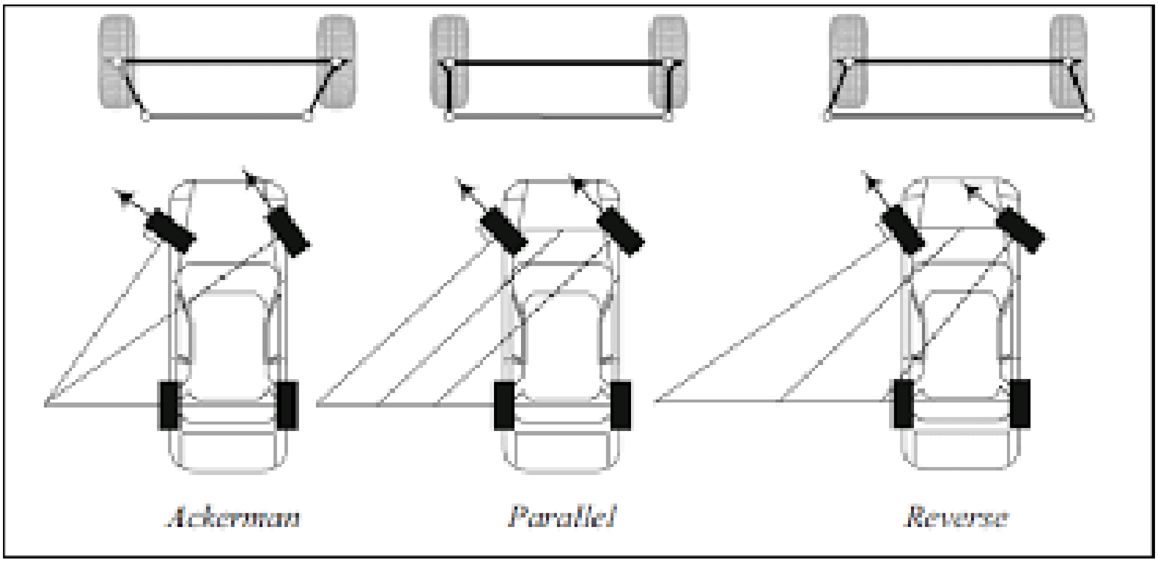
Product Paramenters

Specifications

Model: CHR-GM25-BK370 Permanent magnet dc gear motor											
Voltage:DC6.0V max power 8W Input motor speed 10000RPM											
Special note: Due to the high power and high current of the motor, all customers must have a power supply of 10A or more.											
RATIO	1:4.4	1: 10	1: 20	1: 34	1: 45	1: 57	1: 75	1: 100	1:125	1: 217	1: 478
no-load current(mA)	≤320	≤320	≤320	≤320	≤320	≤320	≤320	≤320	≤320	≤320	≤320
no-load speed(rpm)	2250	1000	490	280	220	175	130	100	80	45	20
rated torque(Kg.cm)	0.3	0.60	1.2	2.0	2.7	3.5	4.5	6.0	max torque 8.0kg.cm (0.8N.m)		
rated torque(N.m)	0.03	0.06	0.12	0.20	0.27	0.34	0.44	0.59			
rated speed(rpm)	1750	790	390	230	175	135	100	80	60	37	18
rated current(A)	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4	≤1.4
stall current(A)	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0	≤8.0
Voltage:DC12.0V max power 26W Input motor speed 20000RPM											
RATIO	1:4.4	1: 10	1: 20	1: 34	1: 45	1: 57	1: 75	1: 100	1:125	1: 217	1: 478
no-load current(mA)	≤600	≤600	≤600	≤600	≤600	≤600	≤600	≤600	≤600	≤600	≤600
no-load speed(rpm)	4500	2000	980	580	440	350	260	200	160	90	40
rated torque(Kg.cm)	0.5	1.1	2.3	3.50	4.50	6.00	7.00	max torque 8.0kg.cm (0.8N.m)			
rated torque(N.m)	0.05	0.10	0.21	0.34	0.44	0.59	0.69				
rated speed (rpm)	3500	1580	780	460	350	275	210	160	130	75	37
rated current (A)	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7	≤2.7
stall current (A)	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0	≤16.0
length of gearbox(L)	19	19.0	19.0	21.5	24.0	24.0	26.5	26.5	29.0	29.0	29.0

**We use DC12V

Steering Mechanic Part



This Future Engineering mission, teams need to build a self-driving car and the steering mechanic determind its direction. In our previous generation robot, we use the "Parallel steering" mechanism in which the wheels angle is equal on both side. With this mechanism, we encounter alot of sliping at both front and back of the car which made it drift and lose its balance. So after our research, we decided to use the "Akerman steering" mechanism. But there are two type of this particular mechanism; "Akerman",which is use globally in any manufactured cars and "Anti-Akerman", which is commonly use in formula series racing cars. "Akerman steering" is when a car is turning, the steering angle is different on both wheel with a tigher angle on the inside and a wider angle on the other side because while turnng the inside wheel turn at a smaller radius while the outside wheel is turning at a larger radius. This eliminate the scrubing(sliding) of the tires and made the turn more smooth. The "Anti-Akerman" on the other hand is the opposite with the outside wheel turnig at a thigher angle generate more grip which is need in a formula car, but it is more difficult to control (the down force generate by formula cars compensate these flaws). In short, the "Akerman steering" mechanism is out best choice and will definitely improve out robot.

Model 3D

Geekservo 360 Degrees

We use Geekservo 2kg 360 Degrees for steering the robot and employ an Ultrasonic Sensor for rotation. This servo is compatible with LEGO, making it easy and convenient to build the robot. We like how you can connect two axles to the dual outputs on this servo so you can power two wheels or gears, or mount the servo securely inside articulated limbs and other contraptions. It's also easy to connect blocks to the sides by poking the studs into the holes.

The gears inside these servos will 'slip' when the blocking load is too high instead of jamming - helping avoid damage to your servos and boards.

The wires are a standard servo pinout -

- Red - positive
- Brown - negative
- Yellow - data

The cable has a fused 1x3 female DuPont socket connector on the end, so it's easy to connect to jumper jerky or 0.1" pitch pin headers.

Specifications:

Temperature Range: -20 ℃ to 60 ℃

Operating Temperature: -10 ℃ to 50 ℃

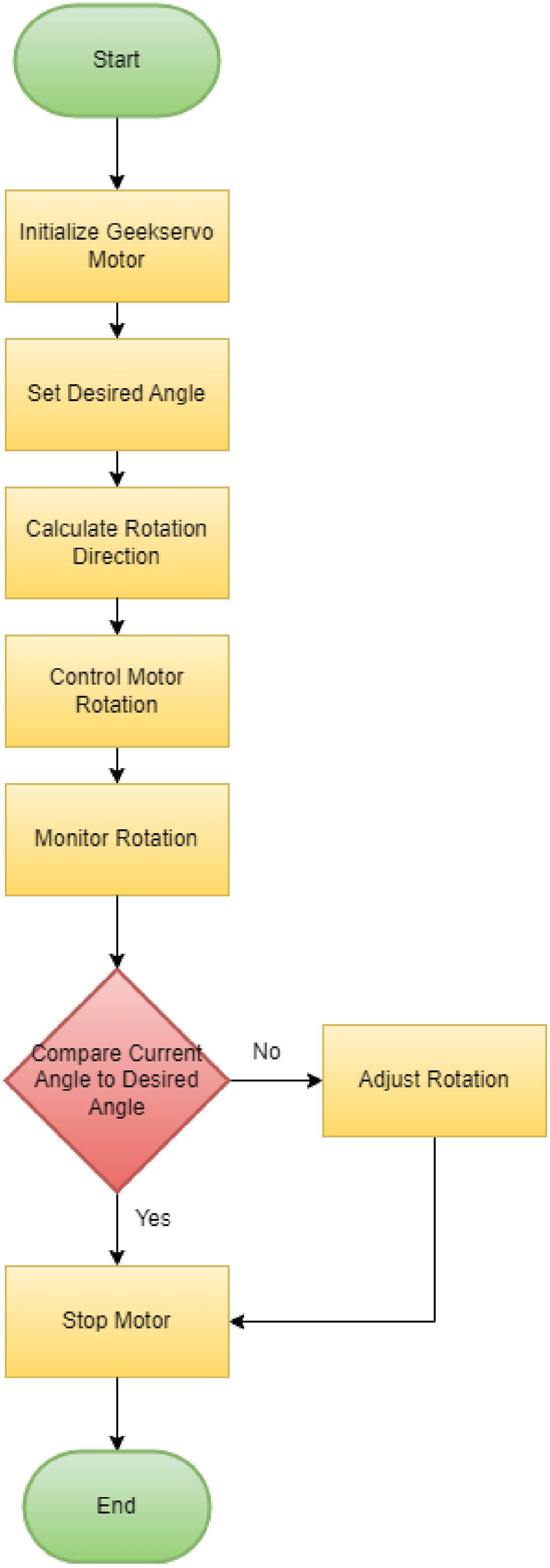
Electrical Voltage: 4.8V ~ 6V (Servo) / 3V (Motor)

Description

- Geekservo 2kg 360 Degrees
- Offers 360-degree dual output axles
- Suitable for middle or large projects
- Compatible with Lego
- Working Voltage: 3.3 - 6 V
- Operating Speed (no load): 0.14sec / 60° (4.8V)

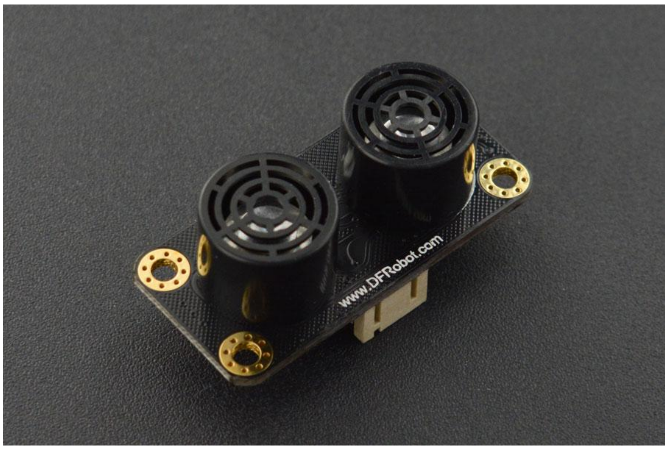
The Geekservo 2kg 360 Degrees Compatible Lego offers 360° dual output axles and is compatible with Lego. It is suitable for middle or large projects and operates on a working voltage range between 3.3 - 6 V voltage.

Flowchart:


Start
Initialize Geekservo Motor
Set Desired Angle
Calculate Rotation Direction
Control Motor Rotation
Monitor Rotation
Compare Current Angle to Desired Angle
No
Adjust Rotation
Yes
Stop Motor
End

2. Energy management and inspection

Ultrasonic Sensor (Gravity: URM09 Ultrasonic Distance)

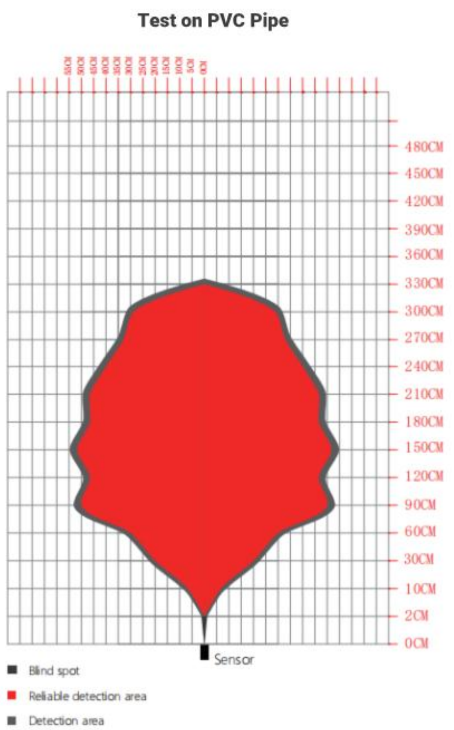


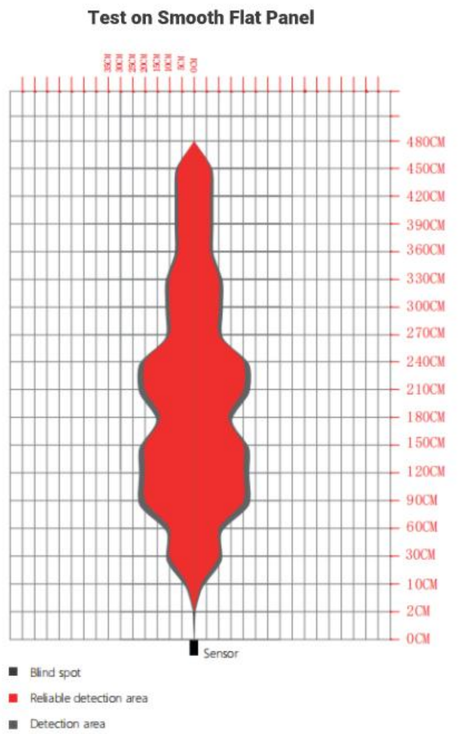
An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back.

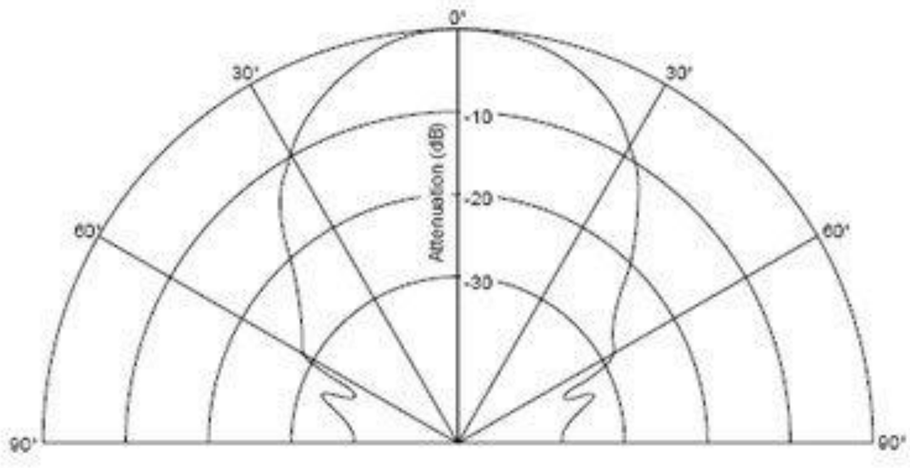
Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

DFRobot URM09 is an ultrasonic sensor specially designed for fast ranging and obstacle avoidance applications. Its measuring frequency can reach up to 30Hz. The sensor adopts built-in temperature compensation and analog output. Meanwhile, it can provide accurate distance measurement within 500cm. The sensor is compatible with Arduino, Raspberry Pi or other main-control

We use the Ultrasonic Sensor (SEN0307) to measure the distance between the robot and the walls. This sensor utilizes analog voltage output and provides accurate







distance measurements within the range of 2-500 cm with a precision of 1 cm and an accuracy of ±1%, making it highly suitable for this competition. boards with 3.3V or 5V logic level.

SPECIFICATION
• Power Supply: 3.3~5.5V DC
• Operating Current: 20mA
• Operating Temperature: -10℃～+70℃
• Measurement Range: 2cm～500cm
• Resolution: 1cm
• Accuracy: 1%
• Frequency: 30Hz Max
• Dimension: 47mm × 22 mm/1.85× 0.87”

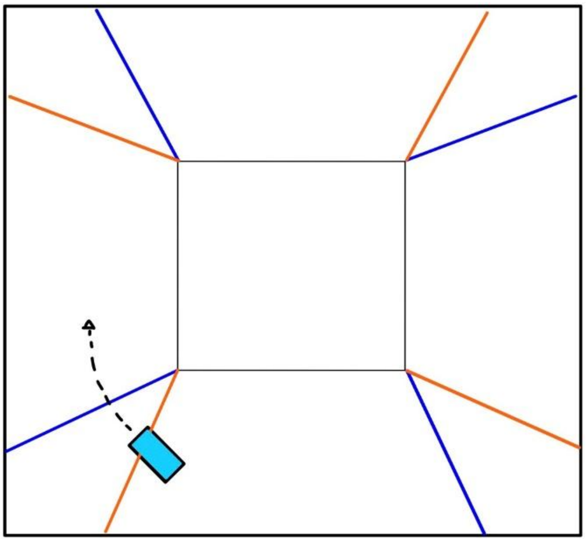
.

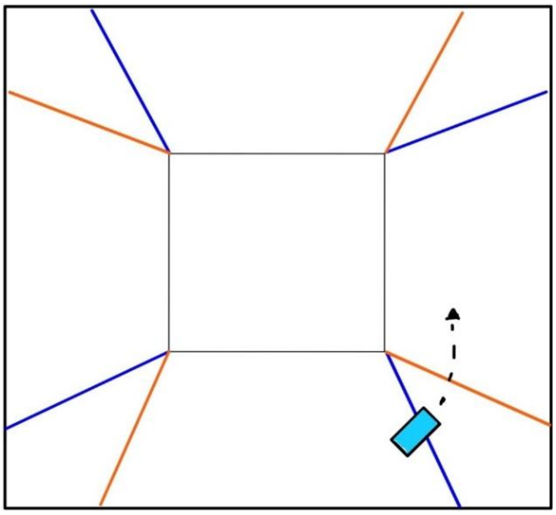
ZX-03 Reflective Infrared Sensor Module

BLUE Light Sensor RED Light Sensor

We use a red sensor to detect the blue line, and a blue sensor to detect the red line. The reason is that optical sensors detect objects more effectively when the emitted light wavelength is different from the color of the target surface. The contrast between the sensor light and the line color increases the reflection difference, resulting in higher accuracy and more reliable detection.

picture turn





Description This module contains a TCRT5000, which integrates an infrared LED and a phototransistor in a single package. When power is supplied, the LED emits infrared light in a wide beam. If an object is placed in front of the sensor, the phototransistor receives the reflected infrared light and allows current to flow. This generates a voltage drop at the S (Vout) pin. The output voltage level depends on the intensity of the reflected infrared light received by the phototransistor — stronger reflections result in higher output voltage, while weaker reflections produce lower voltage.

The TCRT5000 Infrared Reflectance Obstacle Avoidance Line Tracking Sensor is an optical sensor that utilizes infrared (IR) light technology to detect objects and reflective surfaces. It works by emitting infrared light through an IR LED, and then measuring the amount of light reflected back to its built-in phototransistor.

Because of this simple yet effective principle, the TCRT5000 is widely used in robotics and automation projects. One of its most common applications is in line-following robots, where the sensor can distinguish between dark lines and light backgrounds by measuring differences in IR reflectivity. Additionally, it can serve as a short-range proximity sensor, making it suitable for obstacle detection, object counting, and motion tracking systems.

This sensor is popular among students, hobbyists, and engineers alike due to its low cost, compact design, and reliable performance. Its versatility and ease of integration make it an excellent choice for educational projects, competitions, and prototype development.

TCRT5000 Infrared Reflectance Sensor

A compact, budget-friendly sensor that uses an IR LED and phototransistor to detect reflected infrared light.

Perfect for line-following robots, edge detection, proximity sensing, and Arduino DIY projects.

** Features**

Advanced IR Technology – Detects IR reflections, like a third eye for your robot.

Arduino Friendly – Works with 3.3–5V, plug-and-play with Arduino boards.

Versatile Applications – Line tracking, edge detection, proximity sensing, object counting.

Compact & Affordable – Ideal for classrooms, competitions, and prototypes.

Working Principle

The IR LED emits light.

Objects in front reflect IR back to the phototransistor.

Strong reflection → higher Vout

Weak reflection → lower Vout

🛠 Arduino Wiring

Sensor Pin	Arduino Pin
VCC	5V (or 3.3V)
GND	GND
S (Vout)	A0

How to connect and use the Arduino TCRT5000 R3 tracking sensor

```
//Example By ArduinoAll

int ledPin = 13;
```

```
int sensor = A0;
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  //Serial.println("ArduinoAll TEST");
}
void loop() {
  val = analogRead(sensor);  //อ่านค่าจากเซนเซอร์
  Serial.println(val); // แสดงค่าเซนเซอร์ออกทางหน้าจอ
  if (val > 500) { // ค่า 500 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, HIGH); // ไฟ LED ติด
  } else {
    digitalWrite(ledPin, LOW); // ไฟ LED ดับ
  }
  delay(100);
}
```

Open the Serial Monitor to view the values detected by the TCRT5000 line-tracking sensor on the Arduino.

Open MV

The OpenMV Cam is a small and low-power development board with a Cortex-M7 microcontroller supporting MicroPython, a micro-SD card socket and a camera module capable of taking 5MP images - and it's fully supported by Edge Impulse. You'll be able to sample raw data, build models, and deploy trained machine learning models through the studio and the OpenMV IDE.
29 | P a g e

This component is very important for avoiding obstacles. It can detect red and green obstacle from distance to avoid crashing into it. The OpenMV also comes with it's own microcontroller.

Making the robot locate and think faster when see the obstacle. The OpenMV also comes with GLCD screen at the back of it to display what the camera see. This camera can be coded with MicroPython. Additionally this camera wires connect with the sensor shield.

SEN 0253

The Gravity: BNO055 + BMP280 10DOF AHRS is an advanced sensor module that integrates BNO055 and BMP280 on a single board, creating a compact 10 Degrees-of-Freedom (10DOF) sensor for intelligent orientation and environmental sensing.

BNO055 is a small, powerful 9-axis Absolute Orientation Sensor. It is a system-in-package that combines:

A triaxial 14-bit accelerometer

A triaxial 16-bit gyroscope

A triaxial geomagnetic sensor

A 32-bit microcontroller

At only 5.2 x 3.8 x 1.1 mm³, it is significantly smaller than comparable discrete or system-on-board solutions. BNO055 not only provides raw data from the accelerometer, gyroscope, and geomagnetic sensors, but also outputs fused data such as quaternions, Euler angles, and vectors. Its built-in MCU handles complex sensor fusion algorithms, freeing users from manual processing and enabling fast integration into projects like smartphones, wearable devices, and robotics.

BMP280 is an absolute barometric pressure sensor that measures both pressure and temperature, which can be converted to altitude using specific formulas. It uses Bosch's proven piezoresistive pressure technology, offering high accuracy, linearity, long-term stability, and EMC robustness, making it ideal for mobile and embedded applications.

By combining these two sensors, the 10DOF AHRS module provides orientation and environmental sensing in one compact package. The standard Gravity I2C interface simplifies integration, avoiding the complications of multiple sensors from different vendors and saving time for product innovation. This module is ideal for applications such as:

Wearable hardware

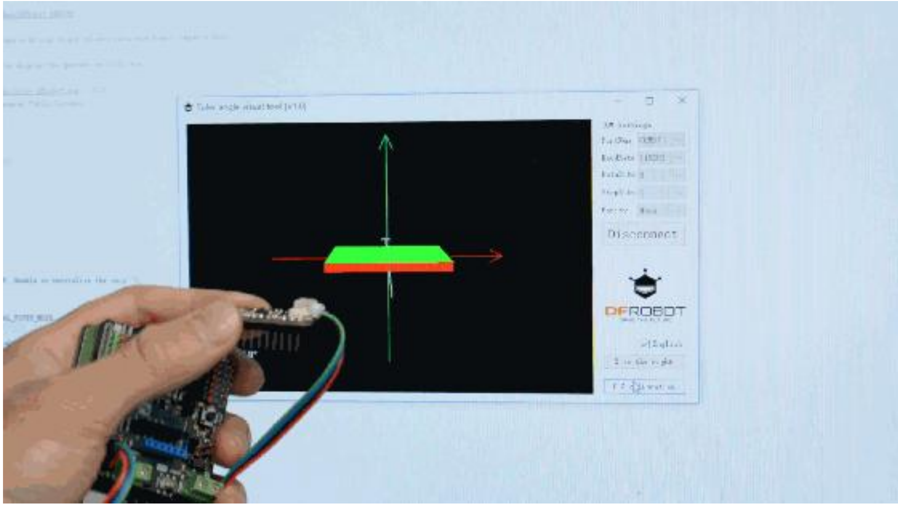
Augmented reality and immersive gaming

Personal health and fitness devices

Indoor navigation and robotics

Any application requiring reliable context awareness

In our project, we previously used the ZX-IMU, but we encountered frequent drift and instability in gyroscope readings. To achieve more stable and reliable orientation data, we switched to the Gravity BNO055 + BMP280 10DOF AHRS, which ensures precise navigation and performance in the WRO Future Engineers 2025 competition.



Sample Code Program Function: Retrieve data from the sensor on the x, y, and z axes, and print it out via the serial port.

```
/*!
  * read_data.ino
  *
  * Download this demo to test read data from bno055
  * Data will print on your serial monitor
  *
  * Product: https://www.dfrobot.com.cn/goods-1860.html
  * Copyright   [DFRobot](https://www.dfrobot.com), 2016
  * Copyright   GNU Lesser General Public License
  *
  * version  V1.0
  * date  07/03/2019
  */
```

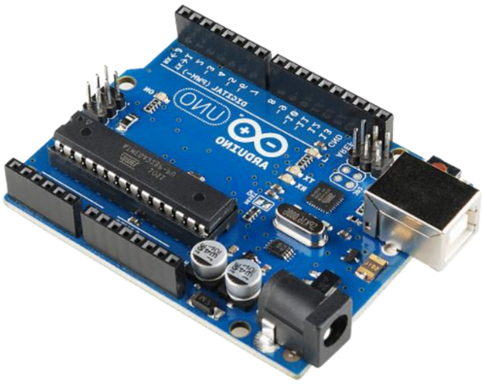
```
#include "DFRobot_BNO055.h"
#include "Wire.h"

typedef DFRobot_BNO055_IIC    BNO;    // ******** use abbreviations instead of full names ********

BNO   bno(&Wire, 0x28);    // input TwoWire interface and IIC address

// show last sensor operate status
void printLastOperateStatus(BNO::eStatus_t eStatus)
{
  switch(eStatus) {
  case BNO::eStatusOK:   Serial.println("everything ok"); break;
  case BNO::eStatusErr:  Serial.println("unknow error"); break;
  case BNO::eStatusErrDeviceNotDetect:   Serial.println("device not detected"); break;
  case BNO::eStatusErrDeviceReadyTimeOut:    Serial.println("device ready time out"); break;
  case BNO::eStatusErrDeviceStatus:    Serial.println("device internal status error"); break;
  default: Serial.println("unknow status"); break;
  }
}

void setup()
{
```

```
  Serial.begin(115200);
  bno.reset();
  while(bno.begin() != BNO::eStatusOK) {
    Serial.println("bno begin faild");
    printLastOperateStatus(bno.lastOperateStatus);
    delay(2000);
  }
  Serial.println("bno begin success");
}

#define printAxisData(sAxis) \
  Serial.print(" x: "); \
  Serial.print(sAxis.x); \
  Serial.print(" y: "); \
  Serial.print(sAxis.y); \
  Serial.print(" z: "); \
  Serial.println(sAxis.z)

void loop()
{
  BNO::sAxisAnalog_t   sAccAnalog, sMagAnalog, sGyrAnalog, sLiaAnalog,
sGrvAnalog;
  BNO::sEulAnalog_t    sEulAnalog;
  BNO::sQuaAnalog_t    sQuaAnalog;
  sAccAnalog = bno.getAxis(BNO::eAxisAcc);    // read acceleration
```

```
  sMagAnalog = bno.getAxis(BNO::eAxisMag);    // read geomagnetic
  sGyrAnalog = bno.getAxis(BNO::eAxisGyr);    // read gyroscope
  sLiaAnalog = bno.getAxis(BNO::eAxisLia);    // read linear acceleration
  sGrvAnalog = bno.getAxis(BNO::eAxisGrv);    // read gravity vector
  sEulAnalog = bno.getEul();                  // read euler angle
  sQuaAnalog = bno.getQua();                  // read quaternion
  Serial.println();
  Serial.println("======== analog data print start ========");
  Serial.print("acc analog: (unit mg)       "); printAxisData(sAccAnalog);
  Serial.print("mag analog: (unit ut)       "); printAxisData(sMagAnalog);
  Serial.print("gyr analog: (unit dps)      "); printAxisData(sGyrAnalog);
  Serial.print("lia analog: (unit mg)       "); printAxisData(sLiaAnalog);
  Serial.print("grv analog: (unit mg)       "); printAxisData(sGrvAnalog);
  Serial.print("eul analog: (unit degree)   "); Serial.print(" head: ");
Serial.print(sEulAnalog.head); Serial.print(" roll: "); Serial.print(sEulAnalog.roll);
Serial.print(" pitch: "); Serial.println(sEulAnalog.pitch);
  Serial.print("qua analog: (no unit)       "); Serial.print(" w: ");
Serial.print(sQuaAnalog.w); printAxisData(sQuaAnalog);
  Serial.println("========  analog data print end  ========");


  delay(1000);
}
```

Controller

Main controller



The Arduino Uno R3 is one of the most widely used and reliable microcontroller boards based on the ATmega328P.
It provides a simple yet powerful platform for building embedded and robotic systems.

Key Features (docs.arduino.cc):

- Microcontroller: ATmega328P (8-bit AVR)
- Clock speed: 16 MHz
- Digital I/O Pins: 14 (6 with PWM output)
- Analog Input Pins: 6

- Memory: 32 KB Flash, 2 KB SRAM, 1 KB EEPROM
- Power: USB (5V) or External (7–12V)
- Onboard reset button, ICSP header, and replaceable MCU chip

Why We Chose Arduino Uno R3

1. Stability and Reliability
   The Uno R3 is known for its robust design and consistent performance, making it highly suitable for competitions where stability is crucial.
2. Full Feature Set for Robotics
   With enough I/O pins, PWM support, and memory, it is capable of handling sensors, motors, and logic required in autonomous robot design.
3. Compliance with WRO Future Engineers Rules
   According to the WRO Future Engineers regulations, teams must use open-source hardware provided through Grammago.
   The Arduino Uno R3, being the most stable

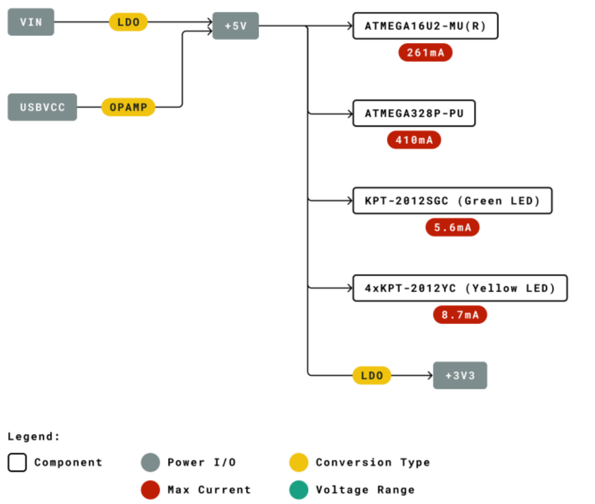
and well-supported open-source controller, was therefore the best choice for our project.

Specifications

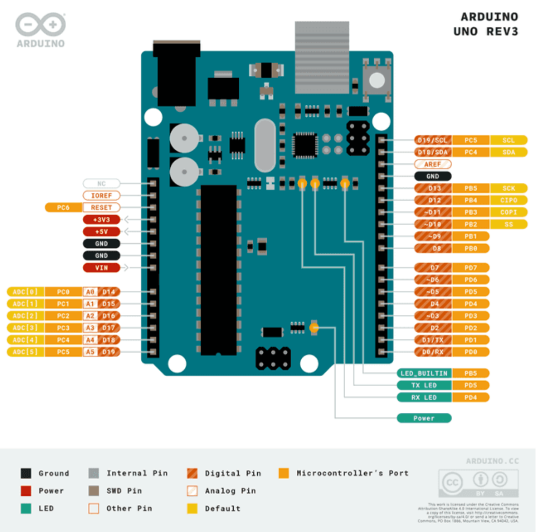
Feature	Specification
Microcontroller	ATmega328P (8-bit AVR)
Operating Voltage	5 V
Input Voltage	7–12 V (recommended), 6–20 V (limits)
Digital I/O Pins	14 (6 support PWM output)
Analog Input Pins	6
DC Current per I/O	20 mA
DC Current for 3.3V	50 mA
Flash Memory	32 KB (0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
LED_BUILTIN	Pin 13
USB Connection	Type-B USB
Other Interfaces	ICSP header, power jack, reset button

Feature	Specification
Dimensions	68.6 mm × 53.4 mm
Weight	~25 g

Power Tree



Pin Out



Expansion Board

L298P DC Motor Drive Shield

The **L298P Motor Driver Shield** is a dual full-bridge driver module based on the **L298P IC**.
It is designed as an **Arduino Uno compatible shield**, making it simple to stack directly on the board without additional wiring.
This shield allows control of **two DC motors** independently (or one stepper motor) with support for both **speed (PWM)** and **direction control**.
It can deliver up to **2A peak per channel**, making it suitable for driving medium-sized DC motors used in robotics and automation projects.

Why We Use the L298P Shield

1. **Seamless Arduino Integration**
   Its shield format makes it directly compatible with the **Arduino Uno R3**, which is our main controller board, ensuring easy wiring and stable operation.
2. **Dual Motor Control**
   The ability to control **two DC motors simultaneously** is essential for our robot's differential drive system.
3. **Reliable and Proven Driver**
   The L298P is one of the most popular and well-documented motor drivers, known for its stability in competitions and projects.
4. **WRO Future Engineers Compliance**
   The rules allow the use of open-source shields and expansion boards. Since the L298P is fully compatible and stable, it became the best choice to power and control our driving motors during the competition.

Product Information

- Input voltage logic portion VD: 5 V

- The driving part of the input voltage VS: VIN Input 6 5 ~ 1 2 V, PWRIN input 4 8 ~ 2 4 V..
- The logic part of the work current I ss: ≤3 6 m A
- The driving part of the work current I o: ≤2 A
- Maximum Power Dissipation: 2 5 W (T = 7 5 ℃)
- A control signal input level: High: 2 3 V≤V in≤5 V Low: -... 0 3 V≤V in≤1 5 V
- Working temperature: - 2 5 ℃ ~ + 1 3 0 ℃
- Drive: Dual high-power H-bridge driver

Feature

- board with L298P motor driver chip, the direct digital IO port with the motherboard (D10 .D11. D12. D13), without cumbersome wiring.
- Onboard beeper (D4), may be provided reversing alarm ringing.
- Convenient motor interface, the motor output can be two routes.
- The two-way Bluetooth interface, can be directly inserted, no wiring.

- out of the D 2, D 3, D 5, D 6, D 7, D 9, D is not occupied by seven digital interface.
- a A 0 - A 5 six analog interface.

Arduino test code:

```
// Arduino Test Code for L298P Motor Shield


// Motor A
int E1 = 10;   // Enable Pin for Motor A (PWM)
int M1 = 12;   // Direction Pin for Motor A


// Motor B
int E2 = 11;   // Enable Pin for Motor B (PWM)
int M2 = 13;   // Direction Pin for Motor B


void setup() {
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
}


void loop() {
  // Forward direction
  for (int value = 0; value <= 255; value += 5) {
    digitalWrite(M1, HIGH);
    digitalWrite(M2, HIGH);
```

```
    analogWrite(E1, value);  // PWM Speed for Motor A
    analogWrite(E2, value);  // PWM Speed for Motor B
    delay(30);
  }
  delay(1000);

  // Backward direction
  for (int value = 0; value <= 255; value += 5) {
    digitalWrite(M1, LOW);
    digitalWrite(M2, LOW);
    analogWrite(E1, value);  // PWM Speed for Motor A
    analogWrite(E2, value);  // PWM Speed for Motor B
    delay(30);
  }
  delay(1000);
}
```

Library [BNO055 Library] [BMP280 Library]

Features BNO055:

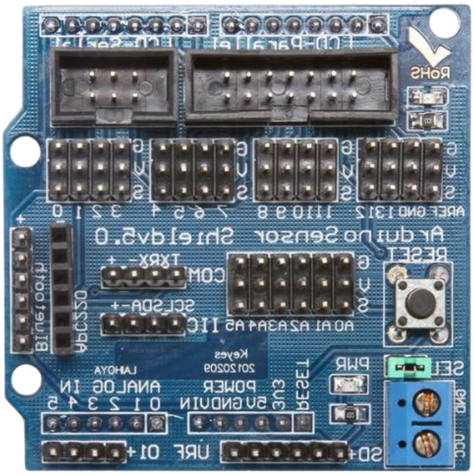
- Outputs fused sensor data: quaternions, euler angles, rotation vector, linear acceleration, gravity, heading.
- 3 sensors in one device: 16-bit gyroscope, 14-bit accelerometer, geomagnetic sensor
- Intelligent Power Management: normal, low power and suspend mode available BMP280 :
- Barometric pressure & Temperature sensor

Specification

Component	Specification
Operating Voltage	3.3V ~ 5V DC
Operating Current	5mA
Interface	Gravity-I2C
BNO055 Accelerometer	- Acceleration ranges: ±2g / ±4g / ±8g / ±16g - Low-pass filter bandwidths: 1kHz ~ <8Hz - Operation modes: normal, suspend, low power, standby, deep suspend - On-chip interrupt control: motion-triggered interrupt
BNO055 Gyroscope	- Ranges: ±125°/s ~ ±2000°/s - Low-pass filter bandwidths: 523Hz ~ 12Hz - Operation modes: normal, fast power up, deep suspend, suspend, advanced power save

Component	Specification
	- On-chip interrupt control: motion-triggered interrupt
BNO055 Geomagnetic Sensor	- Magnetic field range: ±1300μT (x-, y-axis), ±2500μT (z-axis) - Magnetic field resolution: ~0.3μT - Operating modes: low power, regular, enhanced regular, high accuracy - Power modes: normal, sleep, suspend, force
BMP280 Digital Pressure Sensor	- Pressure range: 300 ~ 1100 hPa - Relative accuracy: ±0.12 hPa (±1 m) - Absolute accuracy: ±1 hPa (±8.33 m) - Temperature range: 0℃ ~ 65℃ - Temperature resolution: 0.01℃
Operating Temperature	-40℃ ~ 80℃

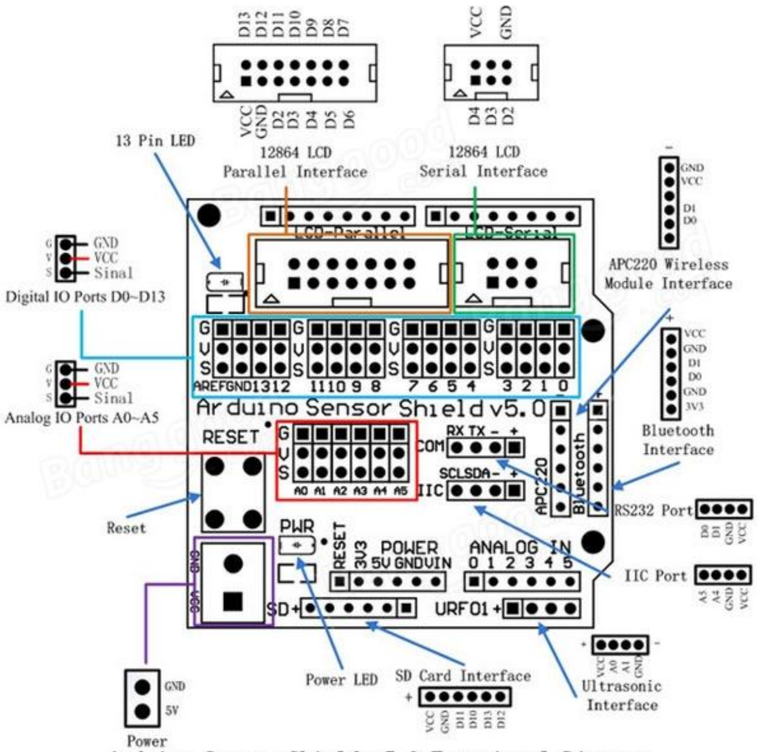
Uno R3 Sensor Shield V5.0 for Arduino



Description The Sensor Shield V5.0 is an expansion board designed to simplify the connection of various sensors, servos, and modules to the Arduino Uno and other compatible boards.
It breaks out Arduino I/O pins into easy-to-use 3-pin headers (VCC, GND, Signal), making it convenient to connect multiple devices without messy wiring or soldering.
This shield supports a wide range of modules such as ultrasonic sensors, IR sensors, servos, relays, and communication modules (Bluetooth, WiFi, etc.).
With clearly labeled pins and dedicated ports for I^2C and UART communication, the Sensor Shield V5.0 is a versatile tool for rapid prototyping and robotics projects.

Why We Use It We chose the Sensor Shield V5.0 because it provides a clean, stable, and organized way to connect

multiple sensors and modules to the Arduino Uno.
For the WRO Future Engineers 2025 competition, wiring stability and reliability are critical during testing and competition rounds.
This shield eliminates the risk of loose jumper wires, saves time during setup, and ensures error-free sensor integration.
It allows us to quickly swap or test different sensors without rewiring the entire circuit, which is especially useful in competitions where adjustments are often needed.



Arduino Sensor Shield v5.0 Functional Diagram

FIREFOX Li-Polymer BATTERRY 20C 1300mAh/11.1V

escription The Firefox Li-Polymer Battery (20C, 1300mAh, 11.1V) is a high-performance rechargeable Li-Po battery designed to provide stable and powerful current output for motors and robotics applications.
With its 20C discharge rate, the battery can deliver bursts of high current, making it suitable for driving motors that require strong torque and consistent performance.
Its compact size and lightweight build make it ideal for robotics projects where both efficiency and power-to-weight ratio are important.

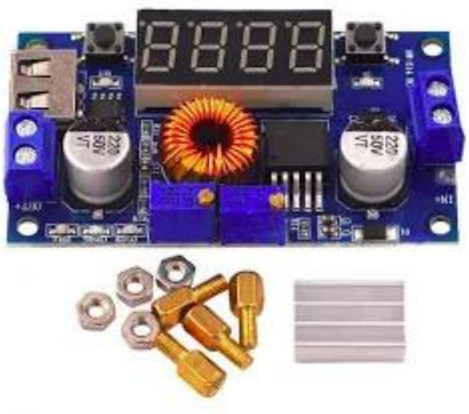
Why We Use It We selected the Firefox Li-Polymer 11.1V 1300mAh battery because it provides enough power to run the Chihai Motor 25-370K at full efficiency.
During testing, we found that this battery delivers strong current without quickly discharging, ensuring longer operation time.

In previous years, we used the Helicox 2200mAh (7.4V / 11.1V) 30C battery, but we faced the problem that it drained too quickly, lasting only 2–3 runs before needing replacement.
To solve this issue, we switched to the Firefox Li-Po, which maintains more stable discharge characteristics and matches our system's power demand better.

This makes the Firefox battery the best balance between power, stability, and endurance, giving our robot reliable performance throughout the WRO Future Engineers 2025 challenge.

Specification	Details
Type	Lithium Polymer (Li-Po), 3 cells, 20C
Voltage	11.1V
Capacity	1200mAh
Length	7 cm
Width	4.2 cm
Thickness	1.6 cm

Step down



This is a step-down DC-DC module. It comes with a status indicator light, a display screen that shows the voltage meter, and self-calibration of the voltage meter. The electrical voltage has an error of 0.05 V, with a measuring range of 0–40 V. We need this step down to show us how long until we need to recharge the battery. Before this, we need to check every 30 minutes. Now, we can use it without worrying about when to recharge.

Technical information

Input voltage: DC 4.0 ~ 38V

Output voltage: DC 1.25V ~ 36V continuously adjustable

Output current: max 5A,

Output power: up to 75W 19 | P a g e

Voltmeter error: ± 0.05V

Measure range: 0 ~ 40V

Conversion efficiency: up to 96%

Load regulation: S (I) ≤ 0.8%

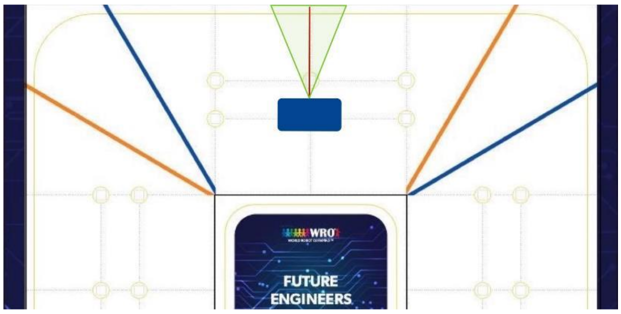
Voltage Regulation: S (u) ≤ 0.8%

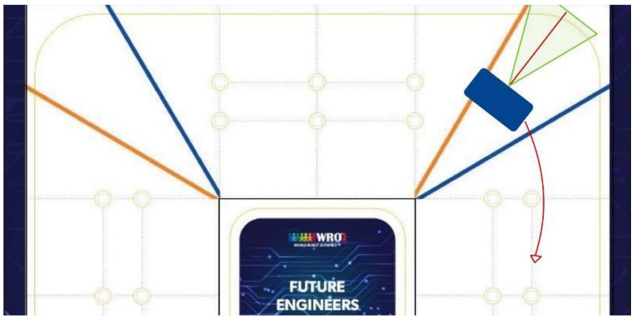
Programming

Qualification Round

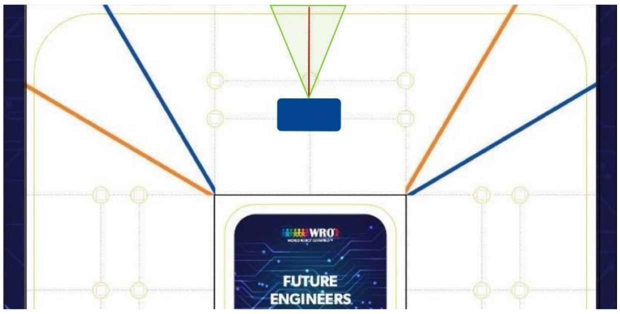
In the qualification round, the robot will use the ultrasonic to measure the distance of the robot from the outer wall. The robot will use the color sensor to check the line color. If the first color is red, the robot will turn right, and if it's blue, the robot will turn left. Every time, the robot will check how many times it has crossed the first line. It will count 12 lines, and after that, it will walk with the timer that we set until the time we set runs out.

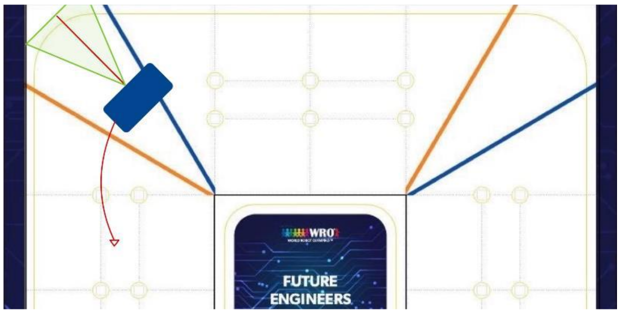
If the first line is red





If the first line is blue





Start When detect the first line

The robot will use the distance of robot from the wall and gyro sensor degree to calculate into steering degree (turning degree) to maintain the distance between the wall with PID (Proportional Integral Derivative)

$$U(t) = k_p e(t) + k_i \int e(t)\, dt + k_\rho \frac{de}{dt}$$

(PID formula)

🚙 **Future Engineer 2025 – Self-Driving Car Robot**

This project is a self-driving robot car designed for the Future Engineer 2025 Open Challenge.
The robot can navigate autonomously using:

1. Line Detection (red/blue lines)
2. Wall Following (PID + ultrasonic servo)
3. Obstacle Avoidance (camera-based)
4. Compass Navigation (BNO055 IMU)

🏗 System Overview

The robot runs a main control loop that continuously processes sensors and makes steering/motor decisions.

```
void loop() {
    camera.handleIncomingData();   // 📷 Process camera data
    getBNO();                      // 🧭 Read compass heading
    line_detection();              // 🎨 Check for colored lines

    int distance_wall = getDistance();  // 📏 Measure wall distance

    // 🎛 Calculate steering using PID controller
    int steering_degree = (1 * x) * compassPID.Run(
        (x * pvYaw) + ((distance_wall - Y)) *
        ((float(Blocks_TURN == 'TURN') - 0.5) * 2)
```

```
    ) * -1;

    // 🚧 Apply obstacle avoidance if blocks detected
    int final_degree = blend_steering_with_avoidance();

    // 🎮 Execute movement commands
    ultra_servo(-pvYaw, Blocks_TURN);  // Rotate sensor to track wall
    motor_and_steer(final_degree);     // Apply steering and speed
}
```

🔧 Hardware Components

The robot is built with the following main components:

Component	Pin	Purpose
BNO055 IMU	I2C	🧭 Compass heading & orientation
OpenMV Camera	Pin 3 (RX)	📷 Object detection & avoidance
Motor Driver	11,12,13	⚡ Speed & direction control
Steering Servo	Pin 9	🎯 Car direction control
Ultrasonic Servo	Pin 8	📐 Sensor rotation for wall tracking

Distance Sensor	Pin A3	📏 Wall distance measurement
Red Line Sensor	Pin A1	🔴 Red line detection
Green Line Sensor	Pin A2	🔵 Blue line detection
Start Button	Pin A0	⭕ Manual start/stop control
Buzzer	Pin 4	🔊 Audio feedback

⚙️ Pin Configuration

```
// Motor Control
#define ENA 11
#define IN1 12
#define IN2 13

// Servo Controls
#define STEER_SRV 9
#define ULTRA_SRV 8

// Sensors
#define ULTRA_PIN A3
#define RED_SEN A1
#define GEEN_SEN A2
#define BUTTON A0
```

```
#define buzzerPin 4


// Communication
#define RX_PIN 3
#define TX_PIN 5
```

🎨 Line Detection

The robot decides turn direction based on line color:

Red line → Turn Right

Blue line → Turn Left

```
void line_detection() {
  int GEEN_value = analogRead(GEEN_SEN);
  int red_value = analogRead(RED_SEN);

  if (GEEN_value < 400 || red_value < 400) {
    simpleBeep();  // 🔊 confirm detection

    int lowest_red_sen = red_value;
    long timer_line = millis();

    while (millis() - timer_line < 100) {
      int red_value = analogRead(RED_SEN);
      if (red_value < lowest_red_sen) {
        lowest_red_sen = red_value;
      }
```

```
        }

        if (lowest_red_sen > 400) {
            // 🔴 Red line → Turn Right
            TURN = 'L';
            Blocks_TURN = 'L';
            x = -1;
            Y = 26;
            compass_offset -= 90;
        } else {
            // 🔵 Blue line → Turn Left
            TURN = 'R';
            Blocks_TURN = 'R';
            x = 1;
            Y = 26;
            compass_offset += 90;
        }
    }
}
```

📐 Wall Following

The ultrasonic sensor rotates with the robot's compass heading, so it always points at the wall.

```
float getDistance() {
    return min(mapf(analogRead(ULTRA_PIN), 0, 1023, 0, 500), 50);
}


void ultra_servo(int degree, char mode_steer) {
    int middle_degree = 0;
    if (mode_steer == 'F') middle_degree = 270;
    else if (mode_steer == 'R') middle_degree = 360;
    else if (mode_steer == 'L') middle_degree = 180;

    Servo_Value = ((max(min(middle_degree + degree, 360), 180)) / 2);
    myservo1.write(Servo_Value);
}
```

🚧 Obstacle Avoidance

The camera detects blocks and calculates how to avoid them.

```
float calculate_avoidance() {
    if (camera.isBlockFound()) {
        BlobData currentBlob = camera.getBlobData();

        if (currentBlob.height > 1.33 * float(currentBlob.width)) {
            found_block = true;
```

```
            float distance = (targetHeight * focalLength * 100) / objectHeight;
            float detected_degree = deltaX * 40 / frameWidth;


            float blockPositionX = distance *
sin(degreesToRadians(detected_degree));
            float blockPositionY = distance *
cos(degreesToRadians(detected_degree)) - 17;


            if (signature == 1) {
                avoidance_degree = min(radiansToDegree(atan2(blockPositionX + 9,
blockPositionY)), -5);
                Blocks_TURN = 'R';
            } else {
                avoidance_degree = max(radiansToDegree(atan2(blockPositionX - 9,
blockPositionY)), 5);
                Blocks_TURN = 'L';
            }
        }
    }
    return avoidance_degree;
}
```

🎛️ Motor & Steering Control

The steering and motor speed adapt dynamically:

```
void motor_and_steer(int degree) {
    degree = min(max(degree, -40), 40);
    steering_servo(degree);

    // 🏎️ Slower speed on turns
    setMotorPercent(map(abs(degree), 0, 45, 50, 60));
}
void setMotorPercent(int sp) {
    sp = constrain(sp, -100, 100);

    if (sp > 0) {
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, map(sp, 0, 100, 0, 255));
    } else if (sp < 0) {
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, HIGH);
        analogWrite(ENA, map(-sp, 0, 100, 0, 255));
    } else {
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 0);
    }
}
```

🏆 Competition Strategy

The robot is designed to complete 3 laps fully autonomously.

Navigation decisions rely on red/blue line detection.

The robot uses wall following for stability.

Obstacle avoidance ensures safe navigation.

The compass keeps turns accurate at intersections.

Set Avoidance & Turn Direction

Red block → turn right: Blocks_TURN = 'R', x = 1

Green block → turn left: Blocks_TURN = 'L', x = -1

avoidance_degree gives the steering correction for the turn.