WDynamics
Hope and Future

TRIdent

User Manual

TRIdent

TRIdent Series

Basic V1.0

■ Overview

This product is a high-performance motor driver board designed to achieve high efficiency and fast response.
It supports BLDC (Brushless DC) motors and PMSM (Permanent Magnet Synchronous Motors), and provides FOC (Field-Oriented Control) capability.

■ Features

MCU

- ◆ STM32F405RGT6(STMicroelectronics)
  A high-performance MCU based on ARM® Cortex®-M4, operating at up to 168 MHz.

BLDC Driver

- ◆ DRV8316(Texas Instruments)
  A 3-phase motor driver with built-in FETs and current-sensing capability.
  Supports up to 40 V and peak current of 8 A.

Angle Sensor

- ◆ TLE5012B-E1000(Infineon)
  GMR-based 15-bit absolute angle sensor.

Interfaces

- ◆ XT30PW (2+2) connector (XT30 power + 2 mm pitch CAN connector)
- ◆ 1.25 mm pitch 3-pin UART connector
- ◆ 1 mm pitch 4-pin ST-LINK connector

Released from May 20, 2026

- 1.27 mm pitch 2-pin socket (for CAN 120 Ω terminator, jumper included)
- Pads for NTC temperature sensor

Indicators

- 6× LED indicators
  1× Power LED (blue) 5× customizable LEDs (blue ×1, green ×1, red ×2, yellow ×1)

Table of Contents

1. Product Introduction

This product is a high-performance BLDC motor driver board designed for high efficiency and fast response.

By implementing an FOC (Field-Oriented Control) program, high-torque and high-precision motor control can be achieved.

2. Technical Specifications

Input Voltage:	24V
Maximum Voltage:	40V
Rated Current:	2A
Maximum Current:	8A
Maximum PWM Frequency:	200kHz
Minimum Operating Temperature:	-40 °C
Maximum Operating Temperature:	+125 °C
Communication:	CAN, UART
Debug IF:	ST-LINK
Angle Sensor:	GMR-based, 15-bit absolute angle
Control Method:	FOC / PWM
Supported Motors:	BLDC / PMSM
Size:	43mm x 41mm

3． Layout Diagram

3．1．Front Side

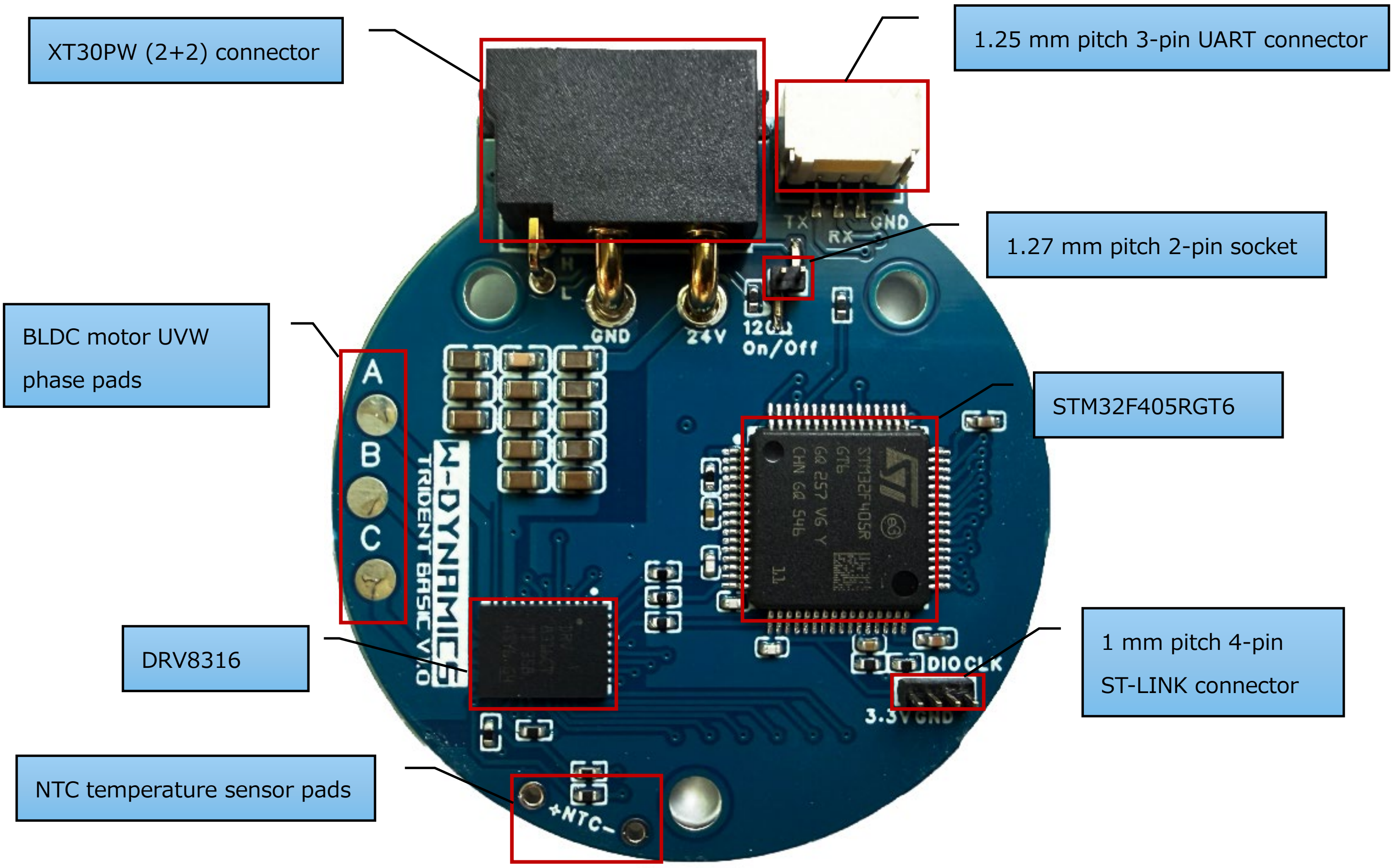

Figure 3.1-1

3. 2. Back Side

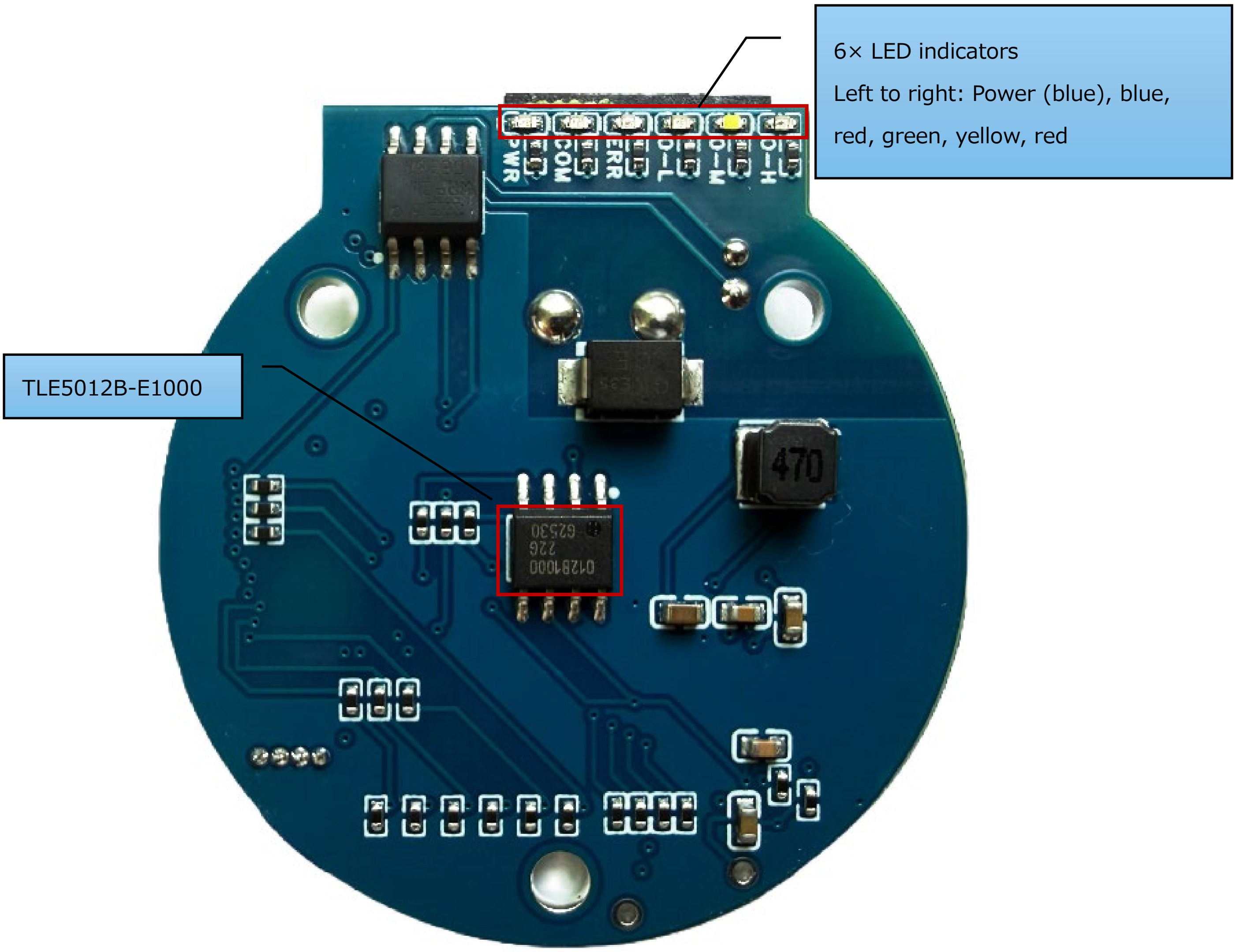

Figure 3.2-1

3. 3. Dimensions

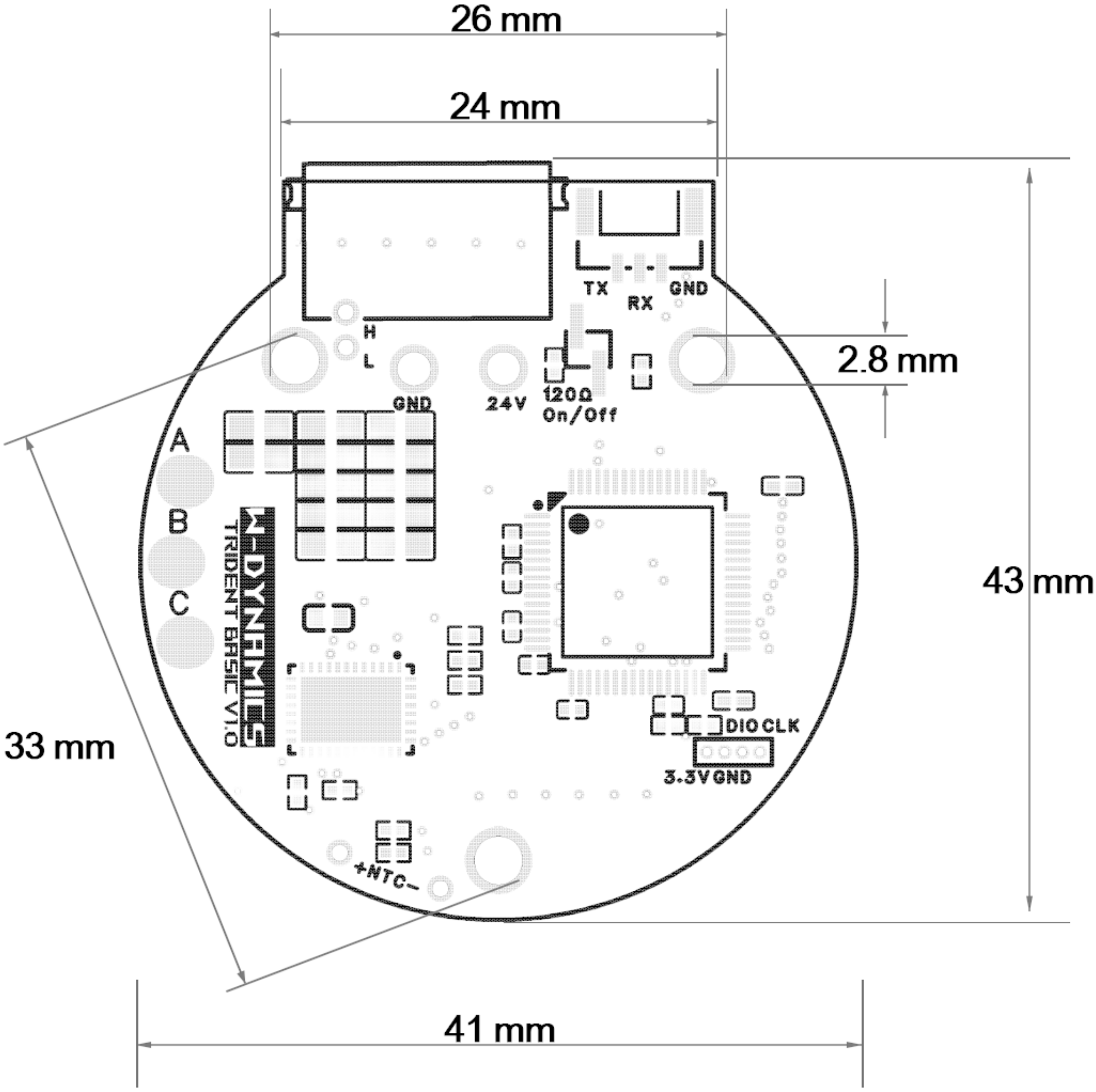

Figure 3.3-1

4. Setup Guidelines

Align the center of the TLE5012B-E1000 angle sensor with the center of the magnet mounted on the motor shaft.
Maintain a spacing of 1–2 mm between the magnet and the sensor.

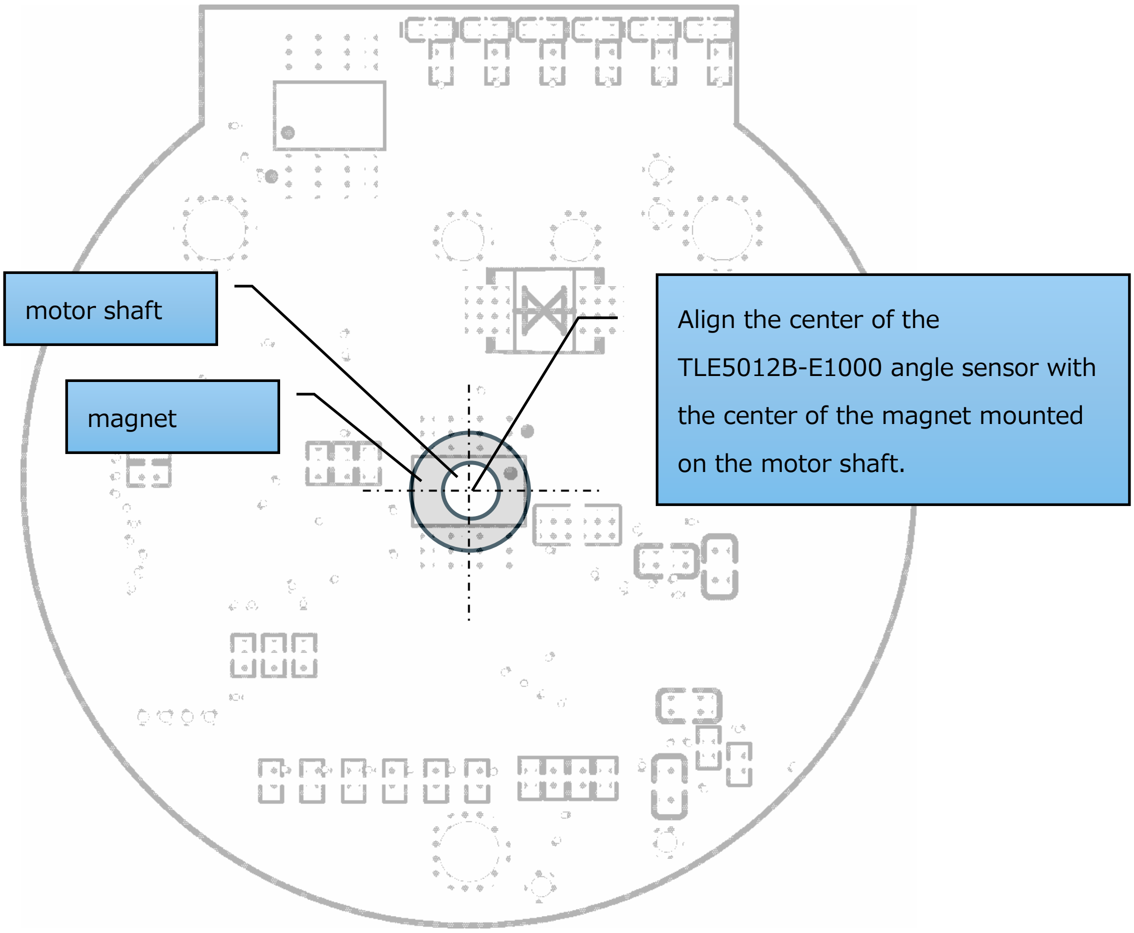

Figure 4-1

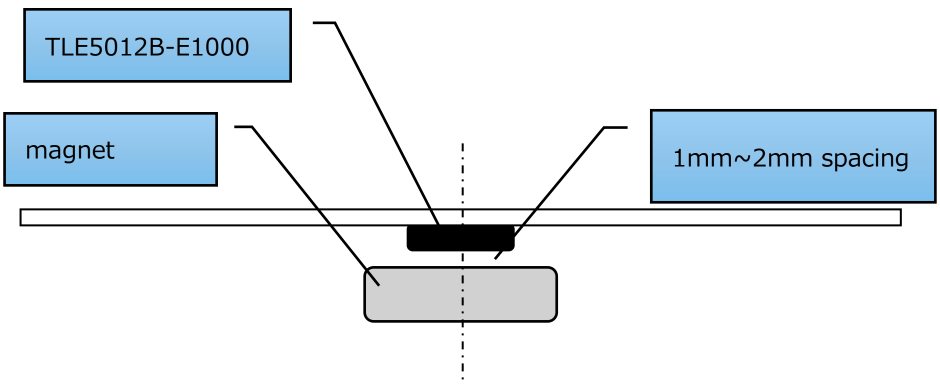

Figure 4-2

5. Simplified Circuit Diagram

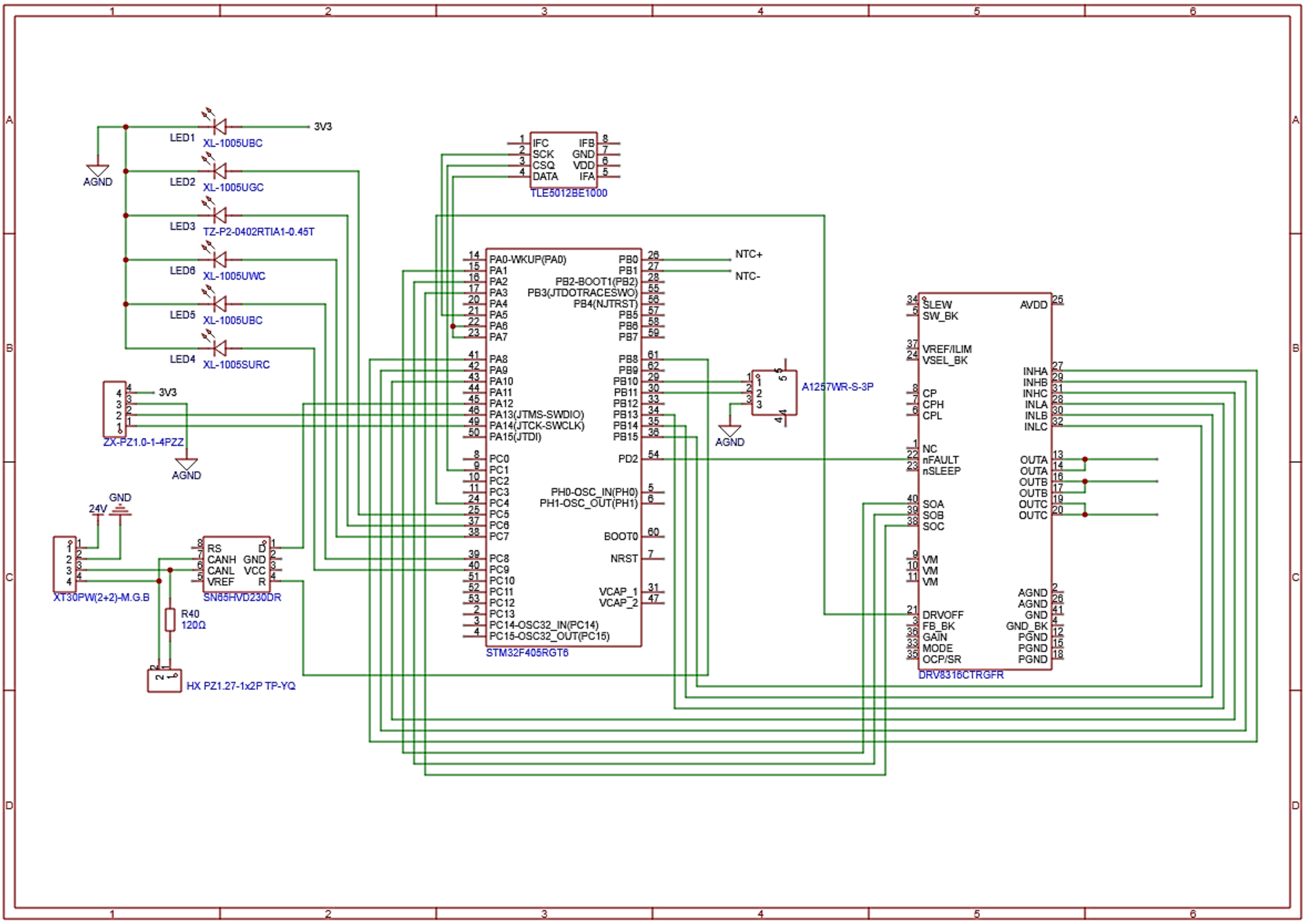

Figure 5 -1

6． Implementation Guidelines

6．1．Simplified Implementation Flow

1)Connect the ST-LINK/V2 debugger to the ST-LINK connector on the driver board.

◆ST-LINK/V2 (STMicroelectronics):
https://www.st.com/ja/development-tools/st-link-v2.html

2)Confirm connection to STM32F405RGT6 using STM32CubeProgrammer.

◆STM32CubeProgrammer:
https://www.st.com/ja/development-tools/stm32cubeprog.html

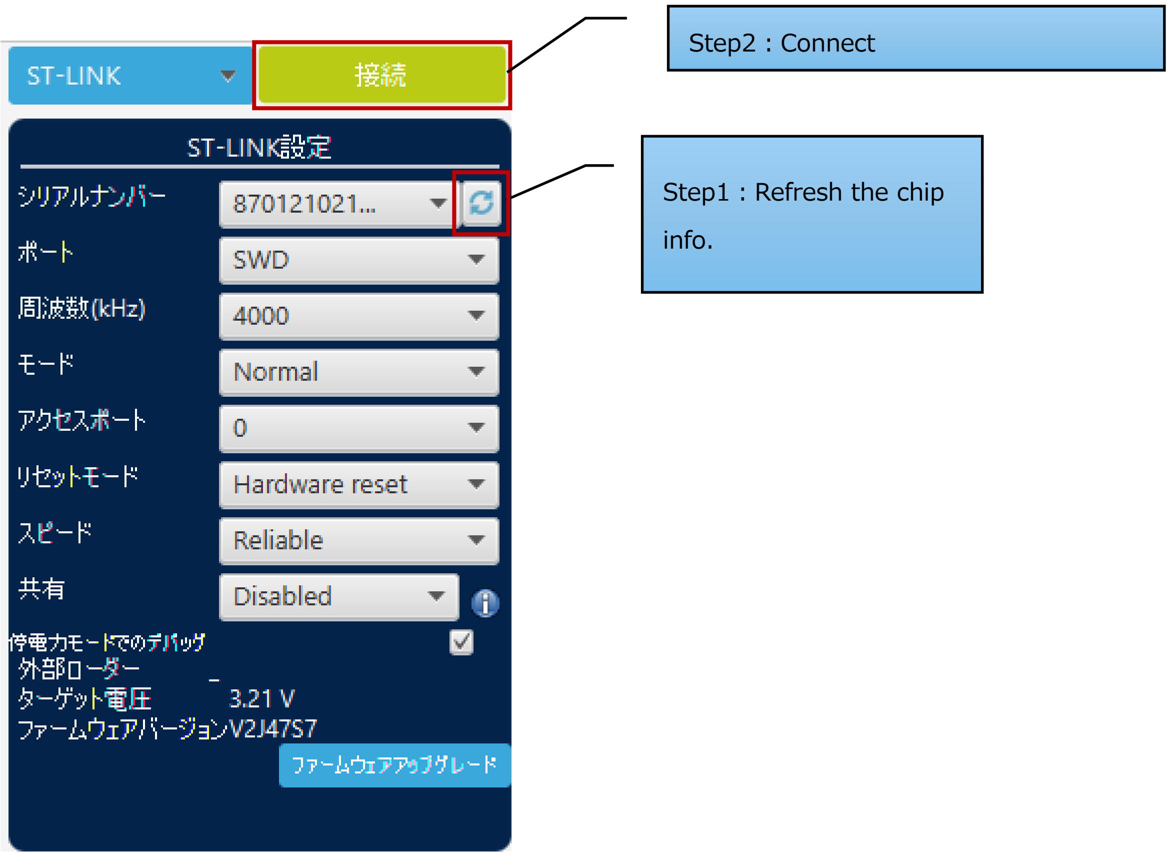

Figure 6 .1-1

Target info will be displayed.

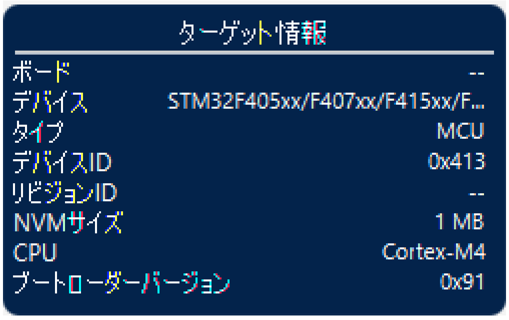

Figure 6 .1-2

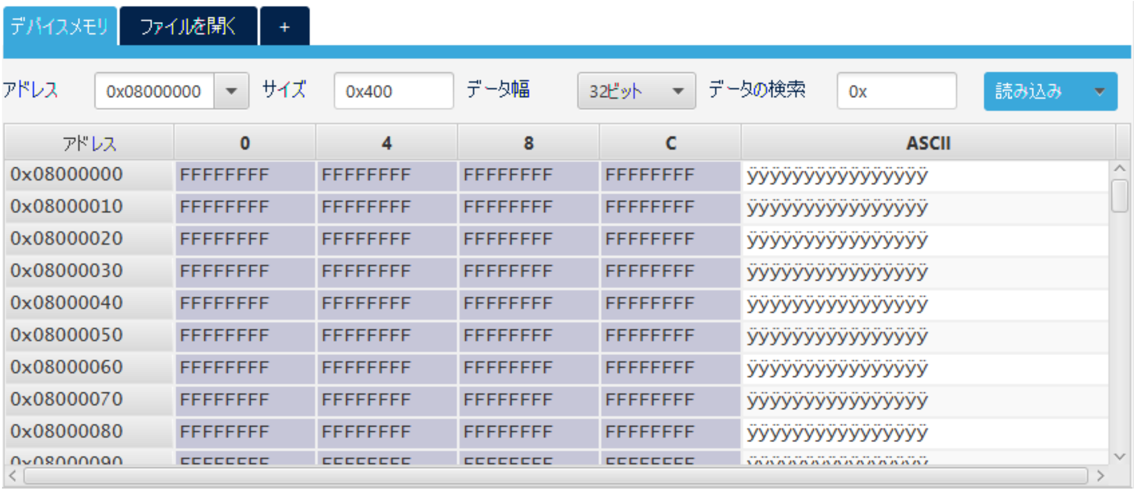

Figure 6 .1-3

3)Use STM32CubeMX to configure initialization settings and generate code.

◆STM32CubeMX:

https://www.st.com/ja/development-tools/stm32cubemx.html

■Recommended Settings :

① PWM Frequency: 20 kHz

SYSCLK : 72 MHz APB2 : 72 MHz

<---------Sample initialization code Start--------->

```
htim1.Instance = TIM1;
htim1.Init.Prescaler = 0;
htim1.Init.CounterMode = TIM_COUNTERMODE_CENTERALIGNED1;
htim1.Init.Period = 1799;
htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim1.Init.RepetitionCounter = 0;
htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
```

<--------- Sample initialization code End--------->

② CAN Baud Rate : 1 Mbps

SYSCLK : 72 MHz APB1 : 36 MHz

<--------- Sample initialization code Start--------->

```
hcan1.Instance = CAN1;
hcan1.Init.Prescaler = 2;
hcan1.Init.Mode = CAN_MODE_NORMAL;
hcan1.Init.SyncJumpWidth = CAN_SJW_1TQ;
hcan1.Init.TimeSeg1 = CAN_BS1_15TQ;
hcan1.Init.TimeSeg2 = CAN_BS2_2TQ;
hcan1.Init.TimeTriggeredMode = DISABLE;
hcan1.Init.AutoBusOff = DISABLE;
hcan1.Init.AutoWakeUp = DISABLE;
hcan1.Init.AutoRetransmission = DISABLE;
hcan1.Init.ReceiveFifoLocked = DISABLE;
hcan1.Init.TransmitFifoPriority = DISABLE;
```

<--------- Sample initialization code End--------->

③ UART Baud Rate: 115200 bps

<--------- Sample initialization code Start--------->

```
huart3.Instance = USART3;
huart3.Init.BaudRate = 115200;
huart3.Init.WordLength = UART_WORDLENGTH_8B;
huart3.Init.StopBits = UART_STOPBITS_1;
huart3.Init.Parity = UART_PARITY_NONE;
huart3.Init.Mode = UART_MODE_TX_RX;
huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart3.Init.OverSampling = UART_OVERSAMPLING_16;
```

<--------- Sample initialization code End--------->

4)Implement motor control source code using STM32CubeIDE.

◆STM32CubeIDE :
https://www.st.com/ja/development-tools/stm32cubeide.html

■Recommended Implementation Steps :
Step1 : Open-loop sine-wave drive (without angle sensor)

★Sample sine-wave generation code:

<--------- Sample code Start--------->

```
// theta：electrical angle、M_PI：Pi
void SetThreePhasePWM(float theta)
{
    // three-phase sine wave create(120 degree)
    float sin_a = sinf(theta);
    float sin_b = sinf(theta - 2.0f*M_PI/3.0f);
    float sin_c = sinf(theta + 2.0f*M_PI/3.0f);

    // Duty create
    float Ua = 0.5f + amplitude *sin_a;
    float Ub = 0.5f + amplitude *sin_b;
    float Uc = 0.5f + amplitude *sin_c;

    // Duty limit
    if (Ua > 1) Ua = 1; if (Ua < 0) Ua = 0;
    if (Ub > 1) Ub = 1; if (Ub < 0) Ub = 0;
    if (Uc > 1) Uc = 1; if (Uc < 0) Uc = 0;

    // Duty output。PWM_PERIOD：PWM Pulse period
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, (uint32_t)(Ua * PWM_PERIOD));
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, (uint32_t)(Ub * PWM_PERIOD));
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_3, (uint32_t)(Uc * PWM_PERIOD));
}
```

<--------- Sample code End--------->

Step2 : Current Control(using angle sensor and ADC)★MOST IMPORTANT★

Acquire currents via ADC to perform Clarke transform

↓

Add electrical angle from angle sensor to perform Park transform

↓

PI control

↓

Inverse Park transform

↓

Generate 3-phase waveform

Step3 : Speed Control

Use speed from angle sensor → perform current control → regulate speed.

Step4 : Position Control

Use position from angle sensor → speed control → current control → position control.

Step5 : Vector Control(FOC)

Optimize dq-axis output parameters for high torque, high efficiency, and smooth rotation.

5)Check output waveforms on pads A/B/C using an oscilloscope.

6)Connect to motor and test using a DC regulated power supply.
Avoid direct battery connection due to risk of exceeding rated current.

6．2．Precautions

★1　Select a motor with similar rated current (2A).
★2　Always perform testing before full operation.
★3　Ensure safety around the rotating motor.
★4　Do not reverse power polarity.
★5　Do not plug/unplug while powered.
★6　Ensure proper heat dissipation.

6．3．FAQ

Q1. Is the input voltage fixed at 24V?
A1. Any voltage ≤ 26 V is acceptable.
Example: A 6-cell Li-ion battery (full charge 25.2 V) is usable.

Q2. My motor’s peak current is 6.5A. Is it unusable?
A2. The board supports up to 8 A peak, so it is usable.
However, the motor must meet the 2 A rated current requirement.

Q3. The mounting holes do not match my motor. What should I do?
A3. Please create a custom mounting frame using a 3D printer.
OEM customization is also available.
Contact: sales@w-dynamics.co.jp

Q4. The oscilloscope shows correct waveforms, but the motor vibrates and does not rotate.
A4. The UVW phase order may be incorrect.Swap any two phases and try again.

Q5. Should the CAN 120Ω terminator jumper always be installed?
A5. If only one driver board is on the CAN line → Install the jumper.
If multiple boards exist → Install the jumper on only one board, remove it from others.