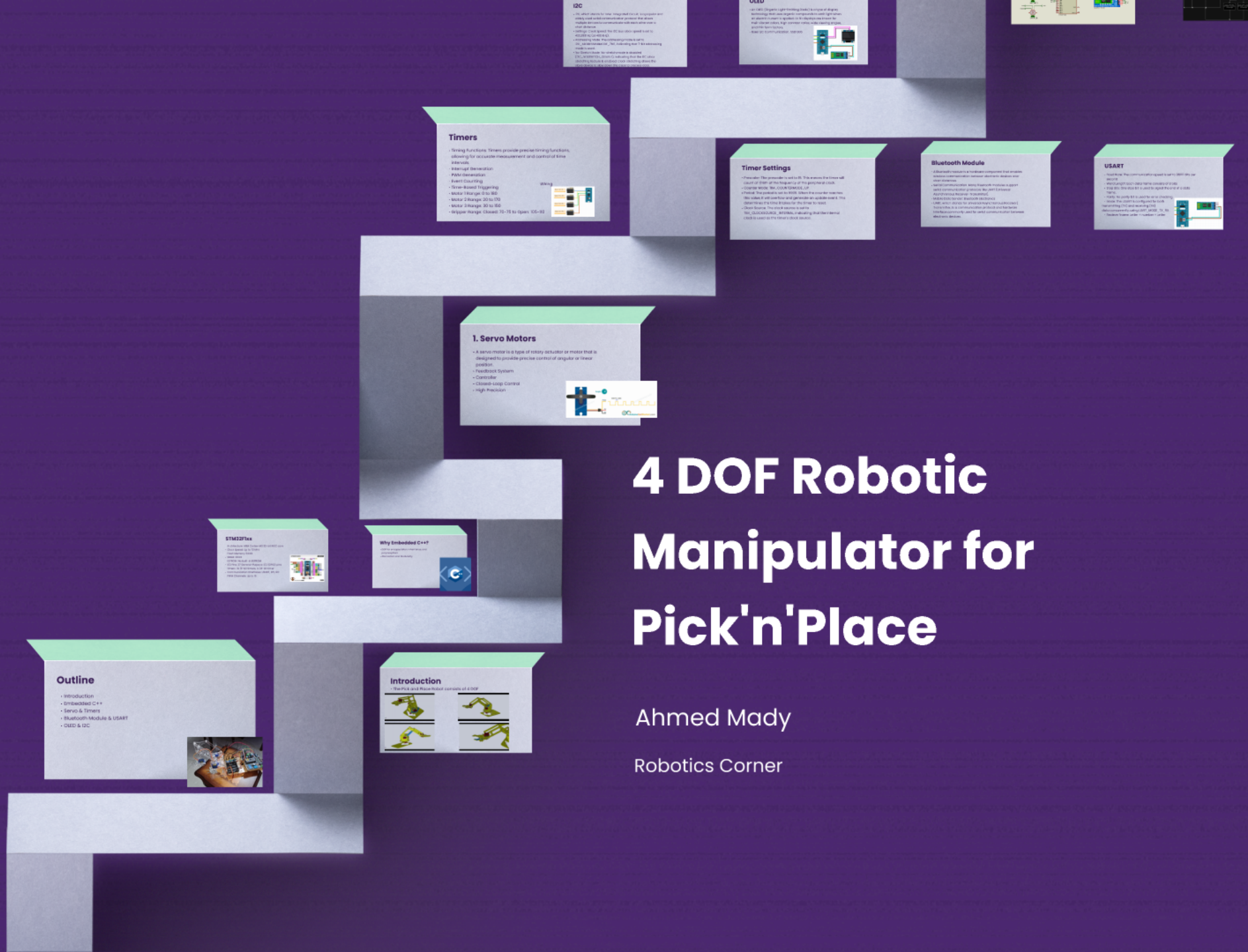


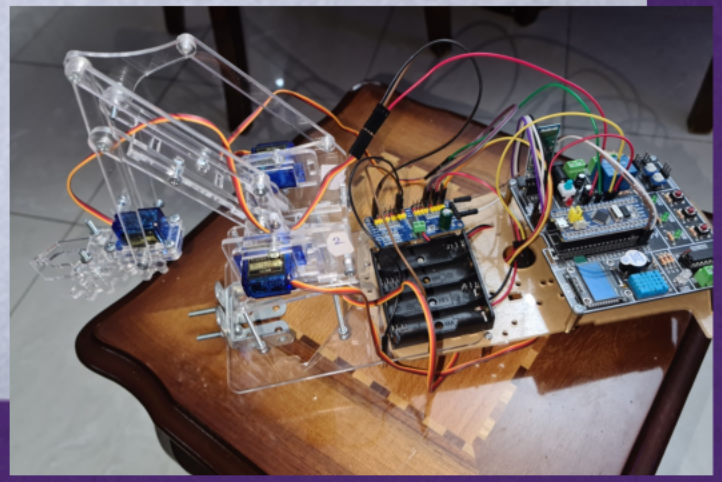
4 DOF Robotic Manipulator for Pick'n'Place

Ahmed Mady

Robotics Corner

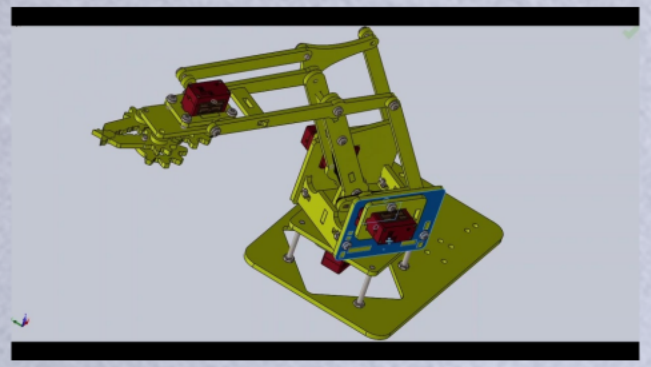
Outline

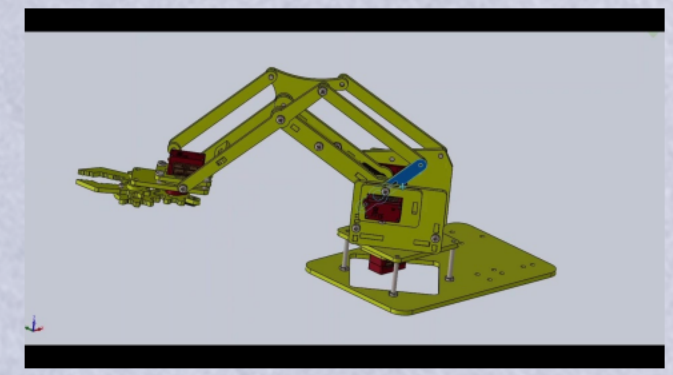
- Introduction
- Embedded C++
- Servo & Timers
- Bluetooth Module & USART
- OLED & I2C

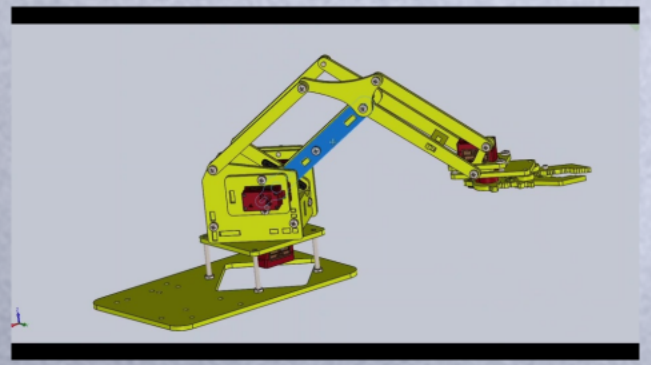


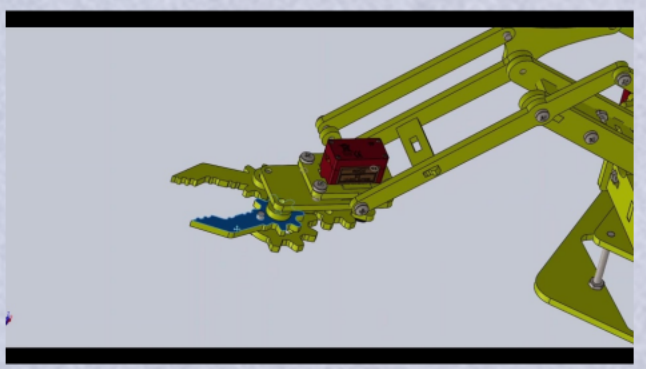
Introduction

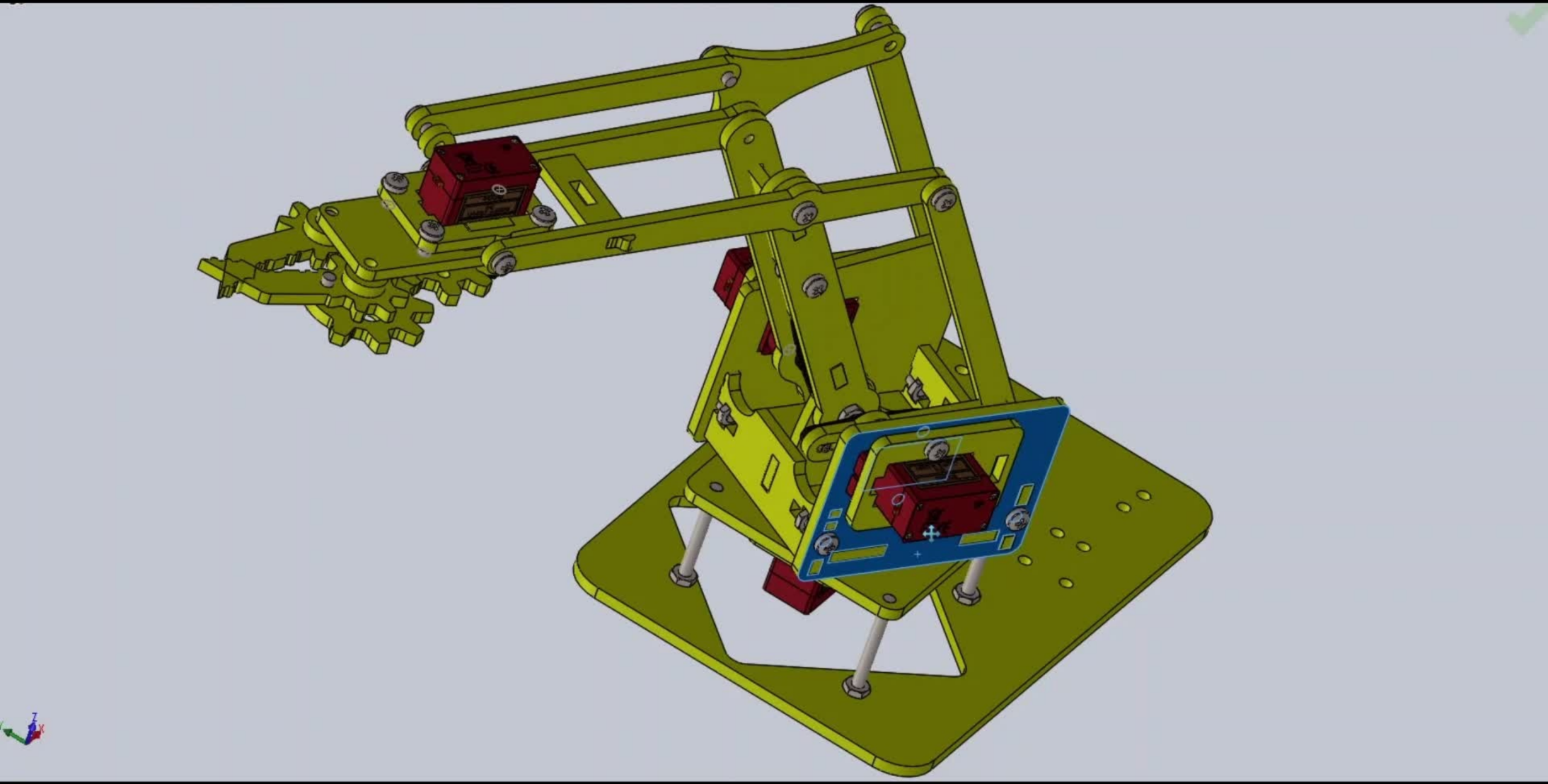
- The Pick and Place Robot consists of 4 DOF

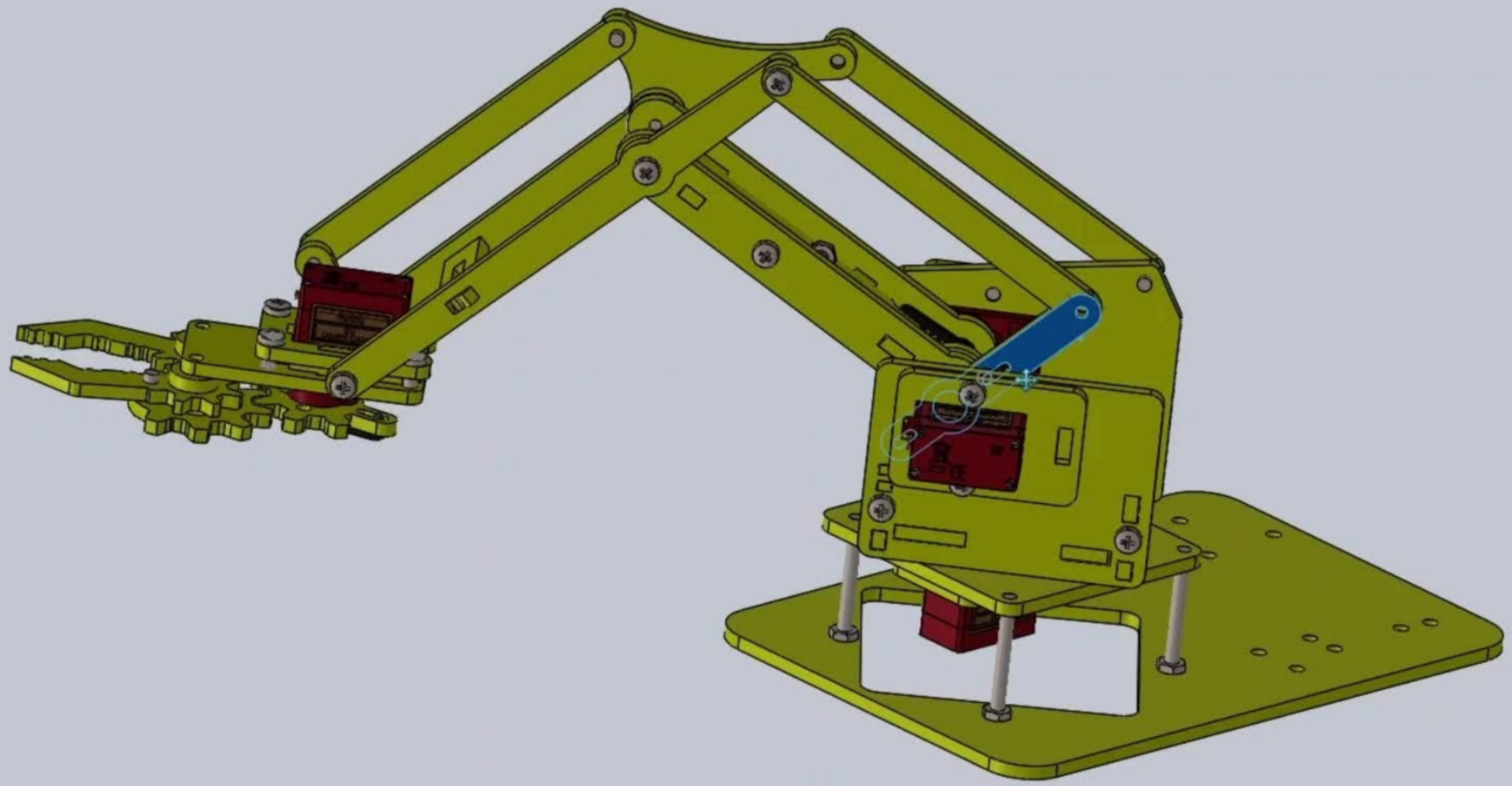


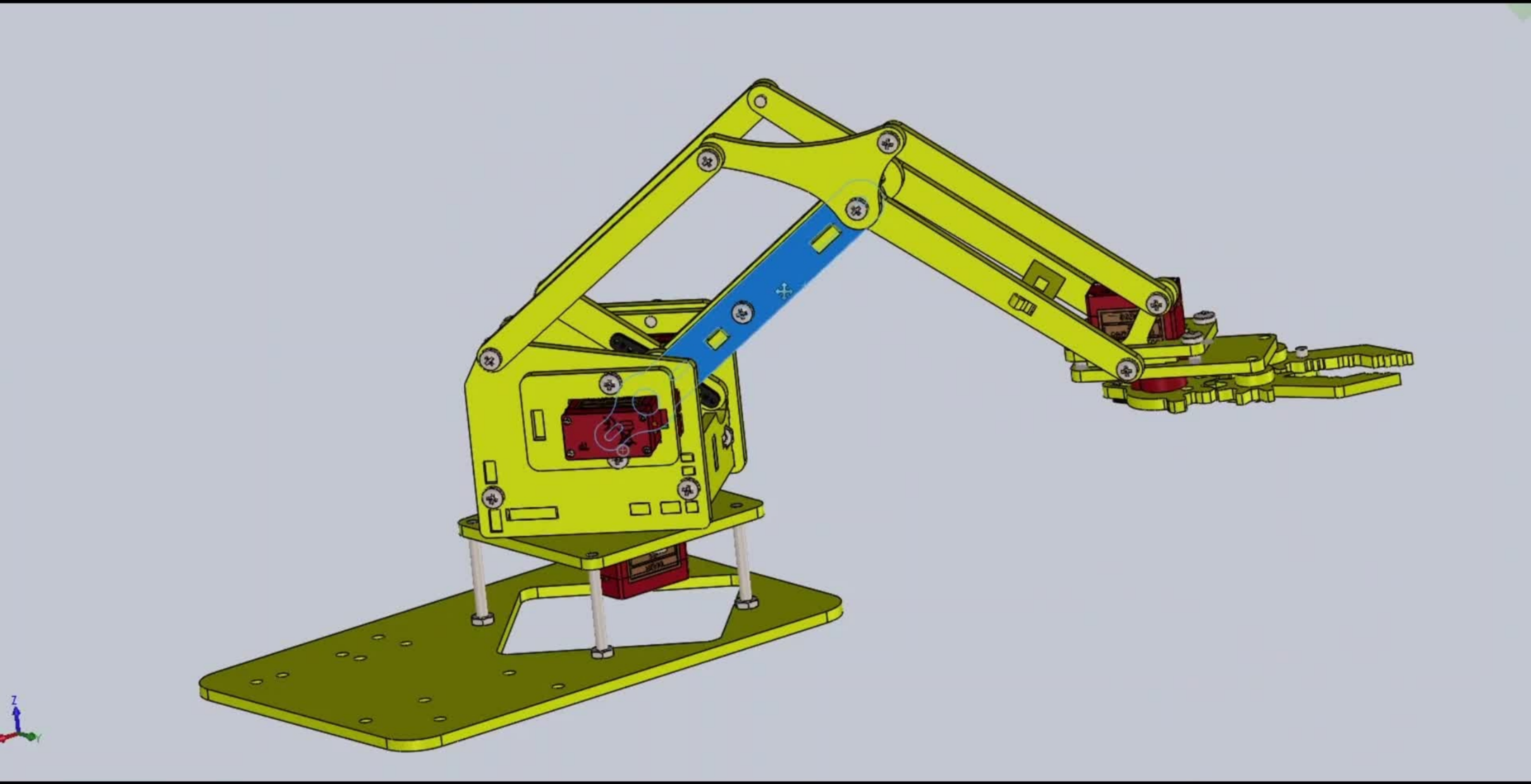


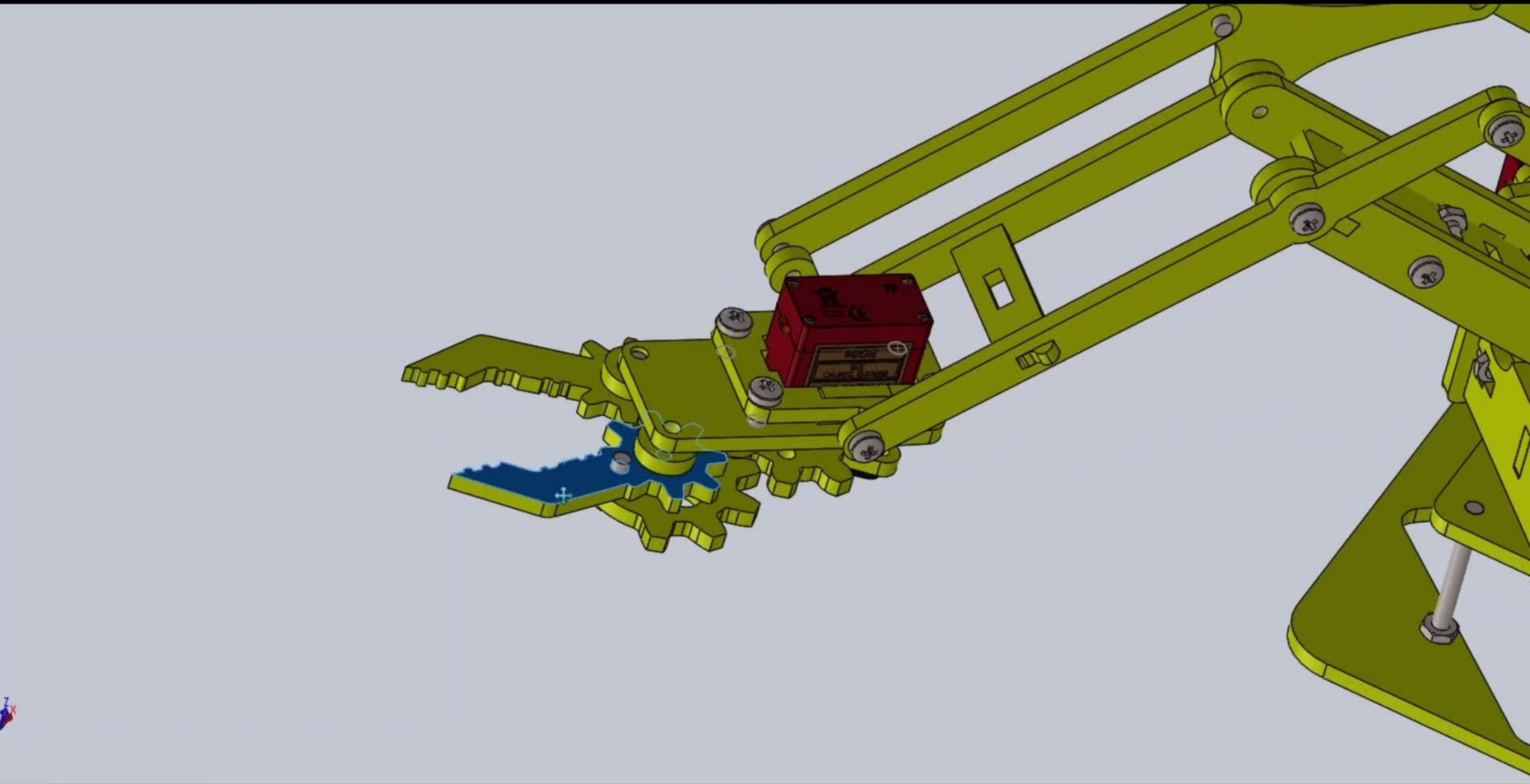






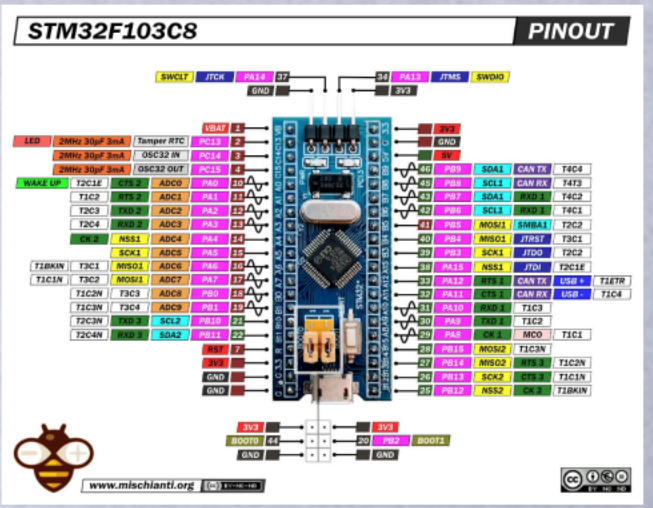

Z
X


Z



STM32F1xx

- Architecture: ARM Cortex-M3 32-bit RISC core
- Clock Speed: Up to 72 MHz
- Flash Memory: 64 KB
- SRAM: 20 KB
- EEPROM: No built-in EEPROM
- I/O Pins: 37 General-Purpose I/O (GPIO) pins
- Timers: 3x 16-bit timers, 1x 24-bit timer
- Communication Interfaces: USART, SPI, I2C
- PWM Channels: Up to 15

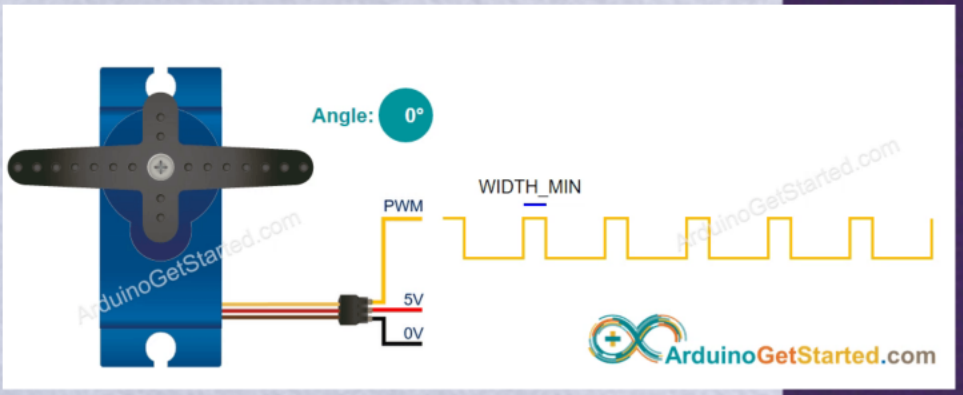


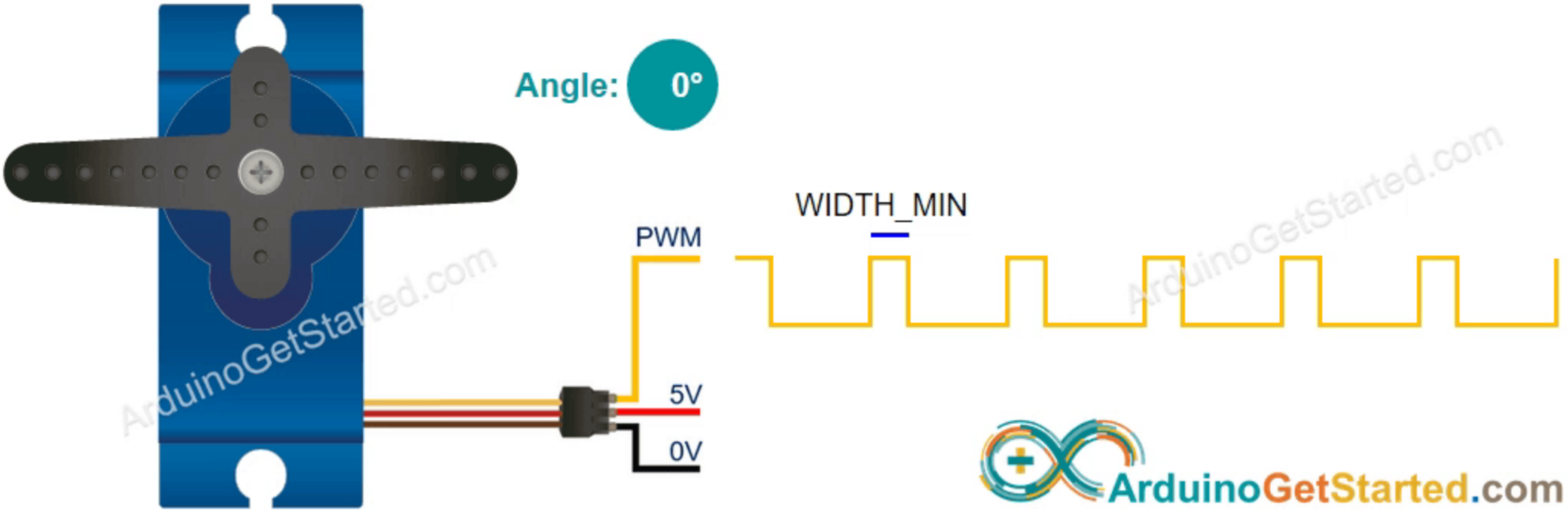
Why Embedded C++?

- OOP for encapsulation, inheritance, and polymorphism
- Abstraction and Modularity

1. Servo Motors

- A servo motor is a type of rotary actuator or motor that is designed to provide precise control of angular or linear position.
- Feedback System
- Controller
- Closed-Loop Control
- High Precision

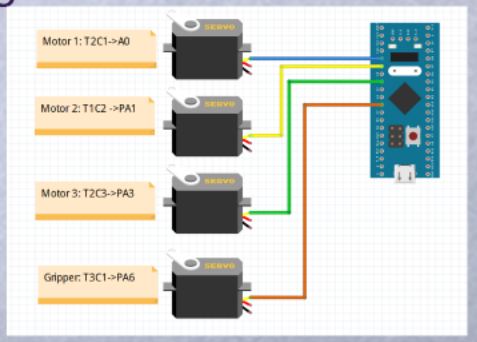



Angle: 0°
WIDTH_MIN
PWM
5V
0V
ArduinoGetStarted.com
ArduinoGetStarted.com
ArduinoGetStarted.com

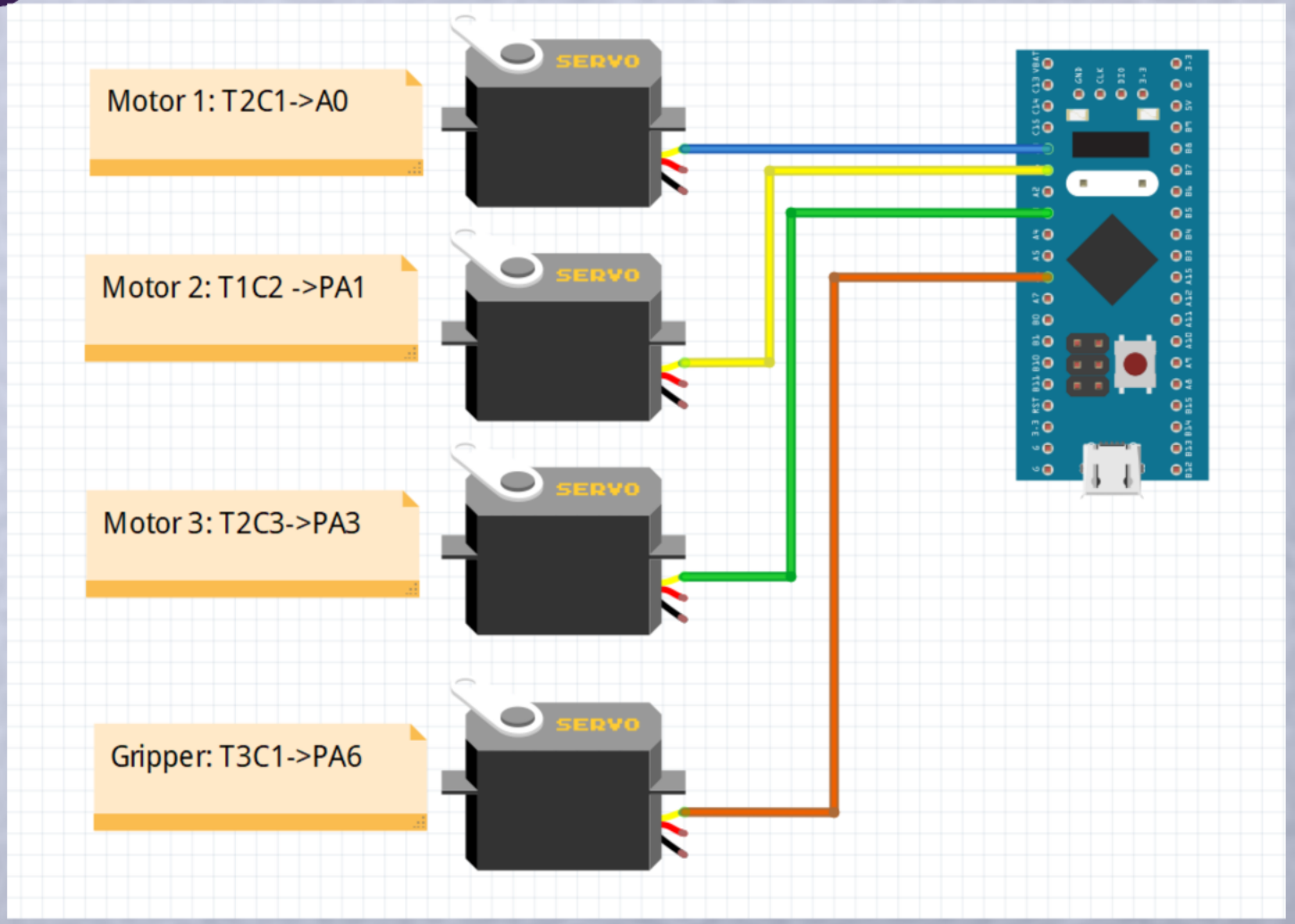
Timers

- Timing Functions: Timers provide precise timing functions, allowing for accurate measurement and control of time intervals.
- Interrupt Generation
- PWM Generation
- Event Counting
- Time-Based Triggering
- Motor 1 Range: 0 to 180
- Motor 2 Range: 20 to 170
- Motor 3 Range: 30 to 150
- Gripper Range: Closed: 70-75 to Open: 105-110

Wiring



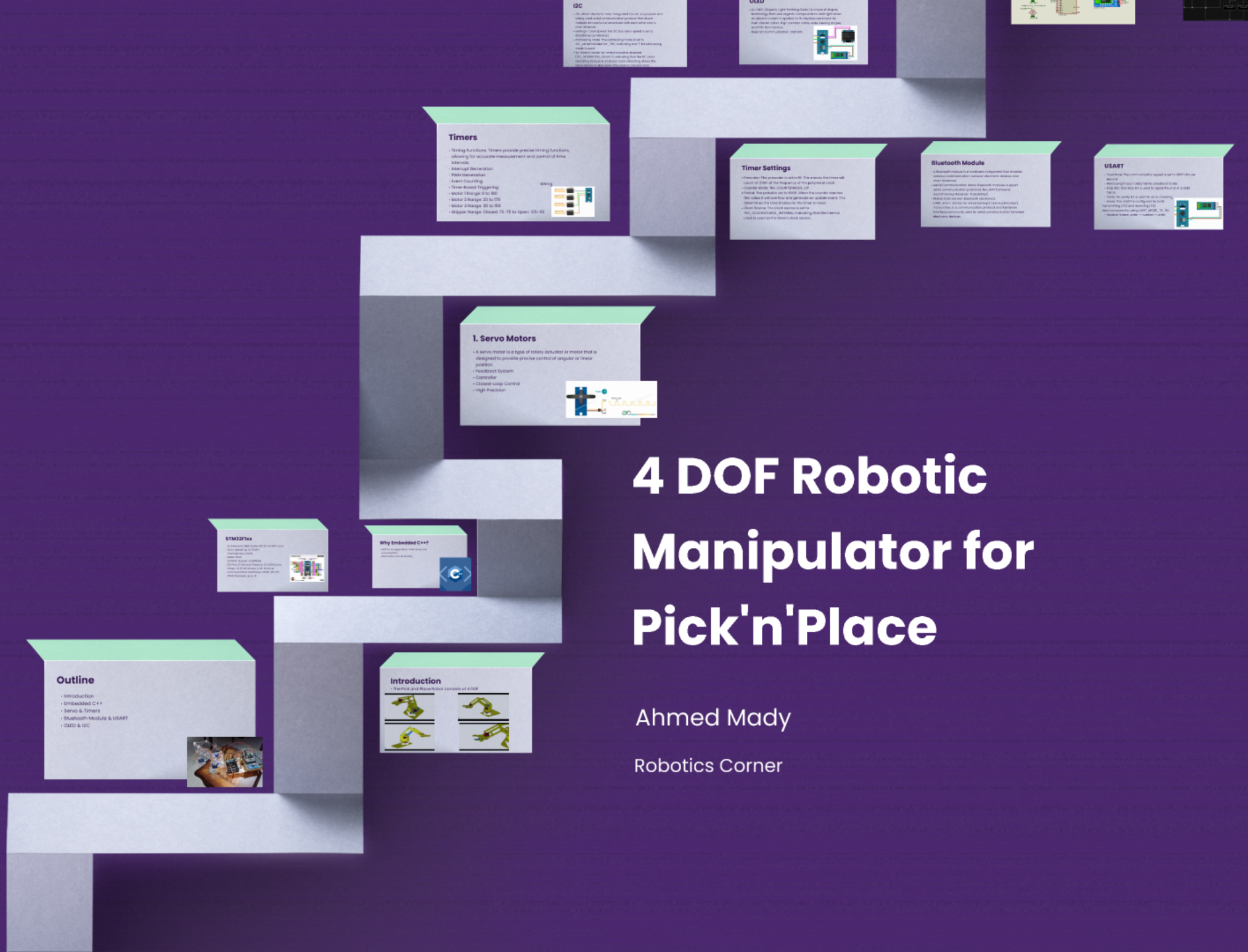
Wiring



105-110

Timer Settings

- Prescaler: The prescaler is set to 15. This means the timer will count at 1/16th of the frequency of the peripheral clock.
- Counter Mode: TIM_COUNTERMODE_UP.
- Period: The period is set to 9999. When the counter reaches this value, it will overflow and generate an update event. This determines the time it takes for the timer to reset.
- Clock Source: The clock source is set to TIM_CLOCKSOURCE_INTERNAL, indicating that the internal clock is used as the timer's clock source.



4 DOF Robotic Manipulator for Pick'n'Place

Ahmed Mady

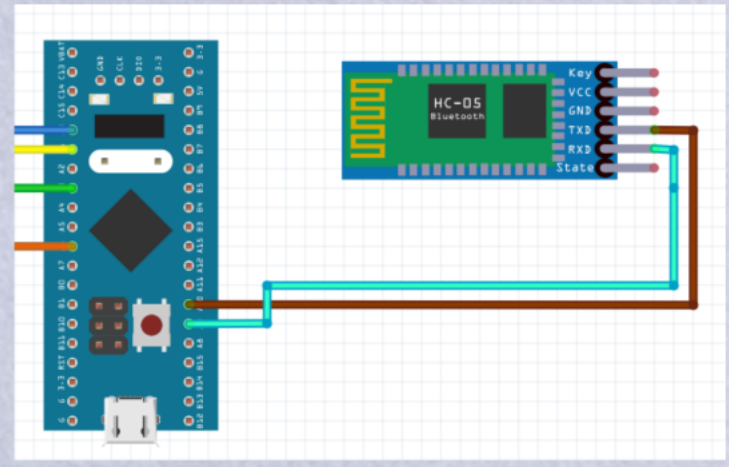
Robotics Corner

Bluetooth Module

- A Bluetooth module is a hardware component that enables wireless communication between electronic devices over short distances.
- Serial Communication: Many Bluetooth modules support serial communication protocols like UART (Universal Asynchronous Receiver-Transmitter).
- Mobile Data Sender: Bluetooth Electronics
- UART, which stands for Universal Asynchronous Receiver/Transmitter, is a communication protocol and hardware interface commonly used for serial communication between electronic devices.

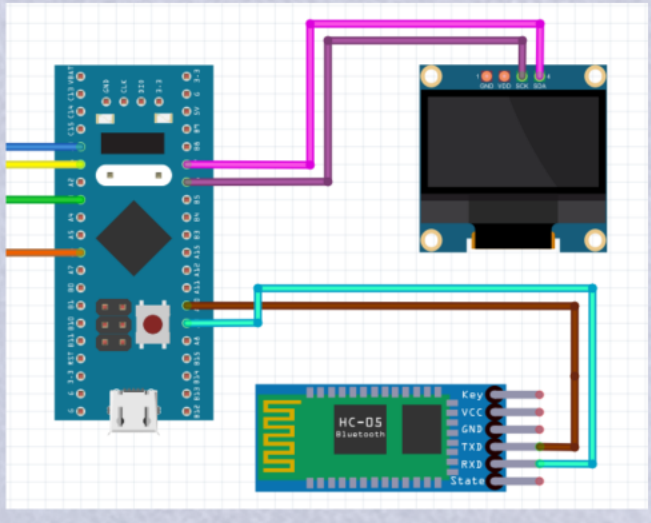
USART

- Baud Rate: The communication speed is set to 9600 bits per second.
- Word Length: Each data frame consists of 8 bits.
- Stop Bits: One stop bit is used to signal the end of a data frame.
- Parity: No parity bit is used for error checking.
- Mode: The USART is configured for both transmitting (TX) and receiving (RX) data concurrently, using UART_MODE_TX_RX.
- Recieve Frame: Letter + number + Letter



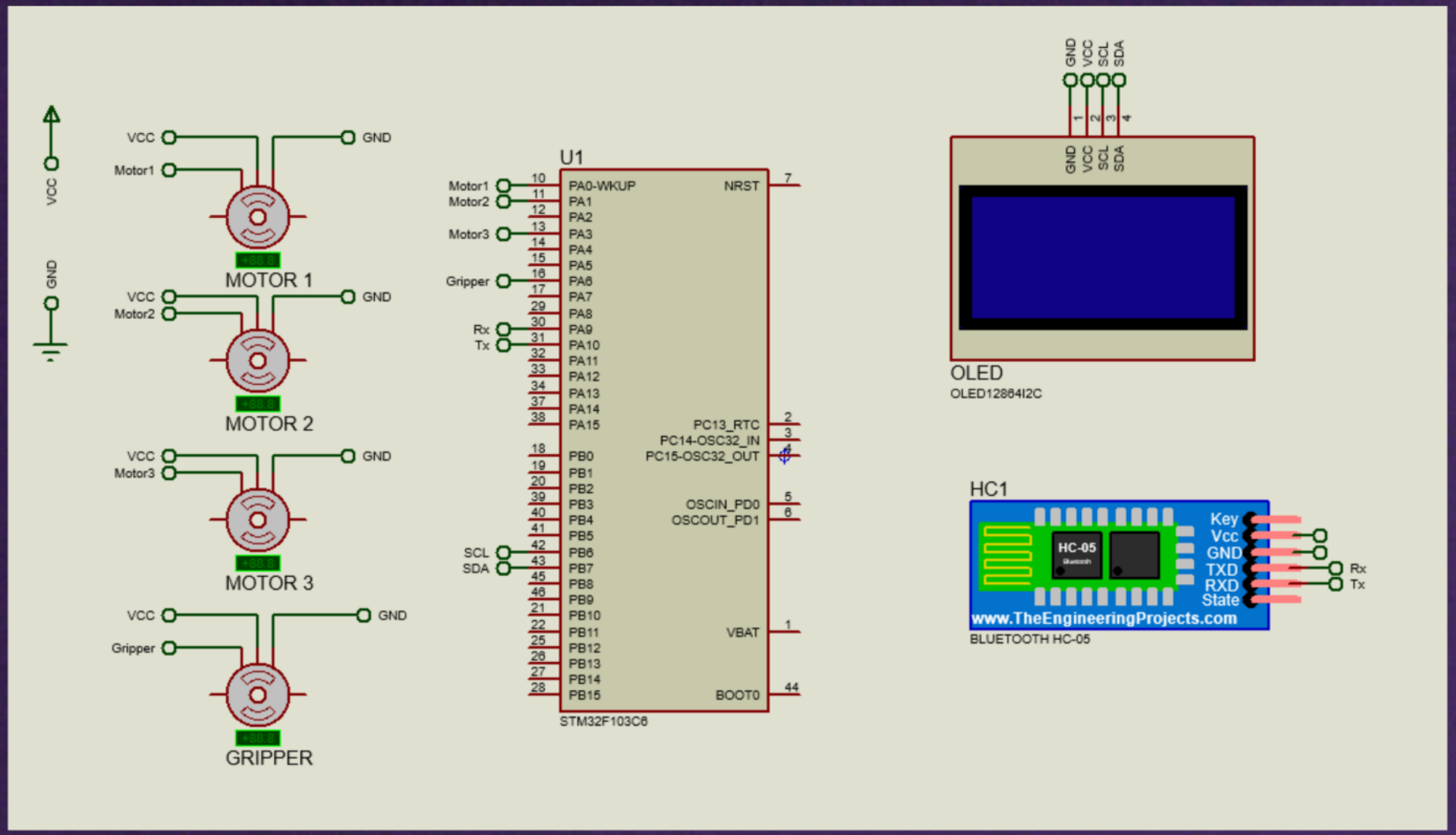
OLED

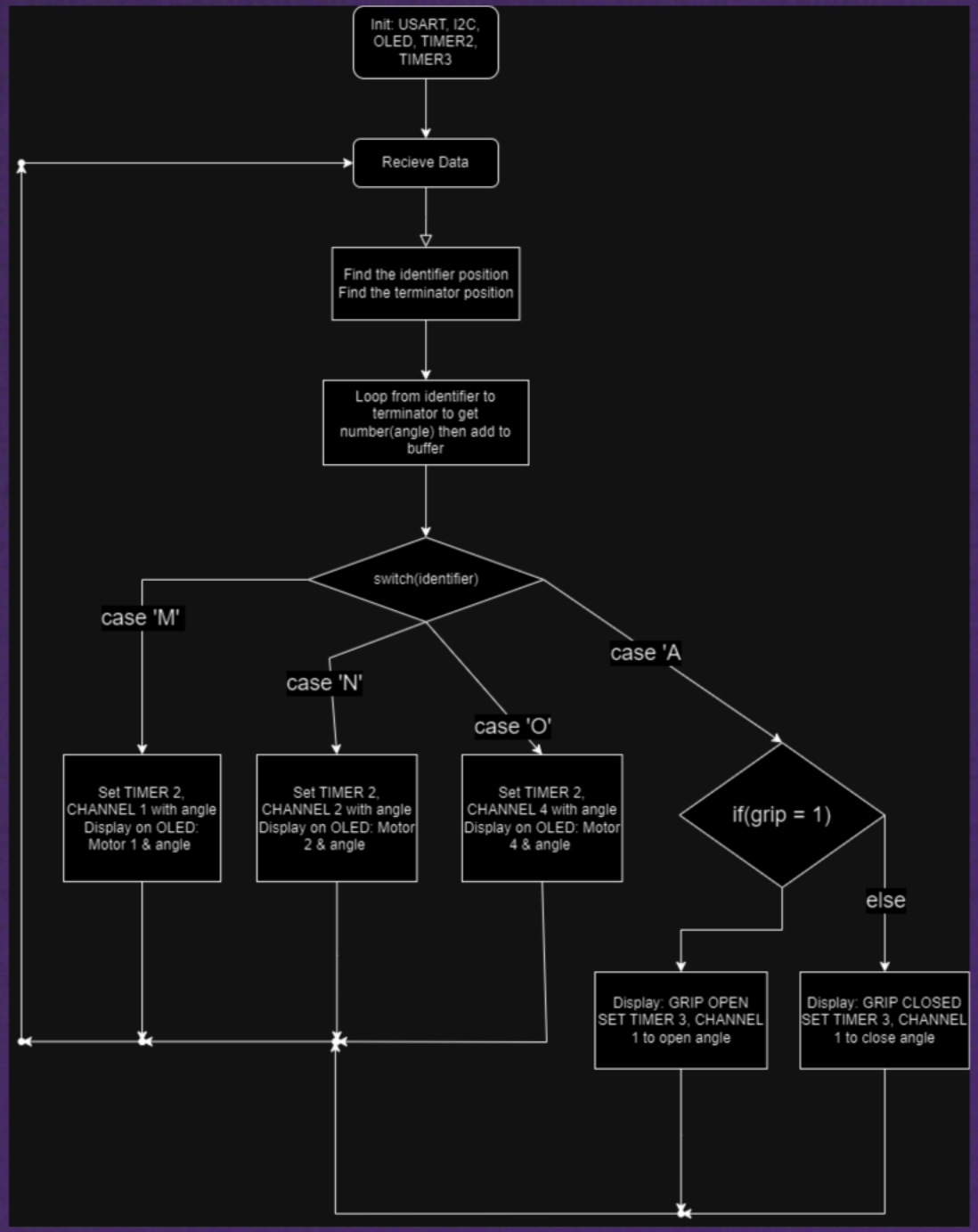
- An OLED (Organic Light-Emitting Diode) is a type of display technology that uses organic compounds to emit light when an electric current is applied. OLED displays are known for their vibrant colors, high contrast ratios, wide viewing angles, and thin form factors.
- Uses I2C Communication, SSD1306



I2C

- I2C, which stands for Inter-Integrated Circuit, is a popular and widely used serial communication protocol that allows multiple devices to communicate with each other over a short distance.
- Settings: Clock Speed: The I2C bus clock speed is set to 400,000 Hz (or 400 kHz).
- Addressing Mode: The addressing mode is set to I2C_ADDRESSINGMODE_7BIT, indicating that 7-bit addressing mode is used.
- No-Stretch Mode: No-stretch mode is disabled (I2C_NOSTRETCH_DISABLE), indicating that the I2C clock stretching feature is enabled. Clock stretching allows the slave device to slow down the clock to process data.


VCC
GND
VCC
Motor1
GND
MOTOR 1
VCC
Motor2
GND
MOTOR 2
VCC
Motor3
GND
MOTOR 3
VCC
Gripper
GND
GRIPPER
U1
Motor1
Motor2
Motor3
Gripper
Rx
Tx
SCL
SDA
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C8
GND
VCC
SCL
SDA
OLED
OLED12864I2C
HC1
HC-05
Key
Vcc
GND
TXD
RXD
State
Rx
Tx
www.TheEngineeringProjects.com
BLUETOOTH HC-05


Init: USART, I2C, OLED, TIMER2, TIMER3
Recieve Data
Find the identifier position
Find the terminator position
Loop from identifier to terminator to get number(angle) then add to buffer
switch(identifier)
case 'M'
case 'N'
case 'O'
case 'A
Set TIMER 2, CHANNEL 1 with angle
Display on OLED: Motor 1 & angle
Set TIMER 2, CHANNEL 2 with angle
Display on OLED: Motor 2 & angle
Set TIMER 2, CHANNEL 4 with angle
Display on OLED: Motor 4 & angle
if(grip = 1)
else
Display: GRIP OPEN
SET TIMER 3, CHANNEL 1 to open angle
Display: GRIP CLOSED
SET TIMER 3, CHANNEL 1 to close angle

Thank YOU