```
// Include Libraries
#include "Arduino.h"
#include "BTHC05.h"
#include "DCMDriverL298.h"
#include "Servo.h"


// Pin Definitions
#define BTHC05_PIN_TXD   11
#define BTHC05_PIN_RXD   10
#define DCMOTORDRIVERL298_PIN_INT1          2
#define DCMOTORDRIVERL298_PIN_ENB6
#define DCMOTORDRIVERL298_PIN_INT2          3
#define DCMOTORDRIVERL298_PIN_ENA5
#define DCMOTORDRIVERL298_PIN_INT3          4
#define DCMOTORDRIVERL298_PIN_INT4          7
#define SERVO360MICRO_PIN_SIG          8



// Global variables and defines

// object initialization
BTHC05 bthc05(BTHC05_PIN_RXD,BTHC05_PIN_TXD);
DCMDriverL298
dcMotorDriverL298(DCMOTORDRIVERL298_PIN_ENA,DCMOTORDRIVERL298_PIN_INT
1,DCMOTORDRIVERL298_PIN_INT2,DCMOTORDRIVERL298_PIN_ENB,DCMOTORDRIV
ERL298_PIN_INT3,DCMOTORDRIVERL298_PIN_INT4);
Servo servo360Micro;


// define vars for testing menu
const int timeout = 10000;       //define timeout of 10 sec
char menuOption = 0;
long time0;

// Setup the essentials for your circuit to work. It runs first every time your circuit is powered
with electricity.
void setup()
{
    // Setup Serial which is useful for debugging
    // Use the Serial Monitor to view printed messages
    Serial.begin(9600);
    while (!Serial) ; // wait for serial port to connect. Needed for native USB
    Serial.println("start");

    bthc05.begin(9600);
```

```
    //This example uses HC-05 Bluetooth to communicate with an Android device.
    //Download bluetooth terminal from google play store,
https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal&hl=en
    //Pair and connect to 'HC-05', the default password for connection is '1234'.
    //You should see this message from your arduino on your android device
    bthc05.println("Bluetooth On....");
    menuOption = menu();

}

// Main logic of your circuit. It defines the interaction between the components you selected.
After setup, it runs over and over again, in an eternal loop.
void loop()
{


    if(menuOption == '1') {
    // HC - 05 Bluetooth Serial Module - Test Code
    String bthc05Str = "";
    //Receive String from bluetooth device
    if (bthc05.available())
    {
    //Read a complete line from bluetooth terminal
    bthc05Str = bthc05.readStringUntil('\n');
    // Print raw data to serial monitor
    Serial.print("BT Raw Data: ");
    Serial.println(bthc05Str);
    }
    //Send sensor data to Bluetooth device
    bthc05.println("PUT YOUR SENSOR DATA HERE");
    }
    else if(menuOption == '2') {
    // L298N Motor Driver with Dual Hobby DC motors - Test Code
    //Start both motors. note that rotation direction is determined by the motors connection to
the driver.
    //You can change the speed by setting a value between 0-255, and set the direction by
changing between 1 and 0.
    dcMotorDriverL298.setMotorA(200,1);
    dcMotorDriverL298.setMotorB(200,0);
    delay(2000);
    //Stop both motors
    dcMotorDriverL298.stopMotors();
    delay(2000);

    }
    else if(menuOption == '3') {
    // Continuous Rotation Micro Servo - FS90R - Test Code
```

```
    // The servo will rotate CW in full speed, CCW in full speed, and will stop  with an interval of 2000 milliseconds (2 seconds)
    servo360Micro.attach(SERVO360MICRO_PIN_SIG);         // 1. attach the servo to correct pin to control it.
    servo360Micro.write(180);  // 2. turns servo CW in full speed. change the value in the brackets (180) to change the speed. As these numbers move closer to 90, the servo will move slower in that direction.
    delay(2000);                          // 3. waits 2000 milliseconds (2 sec). change the value in the brackets (2000) for a longer or shorter delay in milliseconds.
    servo360Micro.write(0);    // 4. turns servo CCW in full speed. change the value in the brackets (0) to change the speed. As these numbers move closer to 90, the servo will move slower in that direction.
    delay(2000);                          // 5. waits 2000 milliseconds (2 sec). change the value in the brackets (2000) for a longer or shorter delay in milliseconds.
    servo360Micro.write(90);    // 6. sending 90 stops the servo
    delay(2000);                          // 7. waits 2000 milliseconds (2 sec). change the value in the brackets (2000) for a longer or shorter delay in milliseconds.
    servo360Micro.detach();                  // 8. release the servo to conserve power. When detached the servo will NOT hold it's position under stress.
    }

    if (millis() - time0 > timeout)
    {
        menuOption = menu();
    }

}



// Menu function for selecting the components to be tested
// Follow serial monitor for instrcutions
char menu()
{

    Serial.println(F("\nWhich component would you like to test?"));
    Serial.println(F("(1) HC - 05 Bluetooth Serial Module"));
    Serial.println(F("(2) L298N Motor Driver with Dual Hobby DC motors"));
    Serial.println(F("(3) Continuous Rotation Micro Servo - FS90R"));
    Serial.println(F("(menu) send anything else or press on board reset button\n"));
    while (!Serial.available());

    // Read data from serial monitor if received
    while (Serial.available())
    {
        char c = Serial.read();
        if (isAlphaNumeric(c))
        {
```

```
        if(c == '1')
                        Serial.println(F("Now Testing HC - 05 Bluetooth Serial Module"));
                else if(c == '2')
                        Serial.println(F("Now Testing L298N Motor Driver with Dual Hobby DC
motors"));
                else if(c == '3')
                        Serial.println(F("Now Testing Continuous Rotation Micro Servo -
FS90R"));
        else
        {
           Serial.println(F("illegal input!"));
           return 0;
        }
        time0 = millis();
        return c;
      }
   }
}
```