

Código
M.C.U.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
```

```
const int sensorPin = 2; // Pin del sensor de velocidad
const int pin3_motor = 4;
const int pin4_motor = 3;
const int ENA = 5;
```

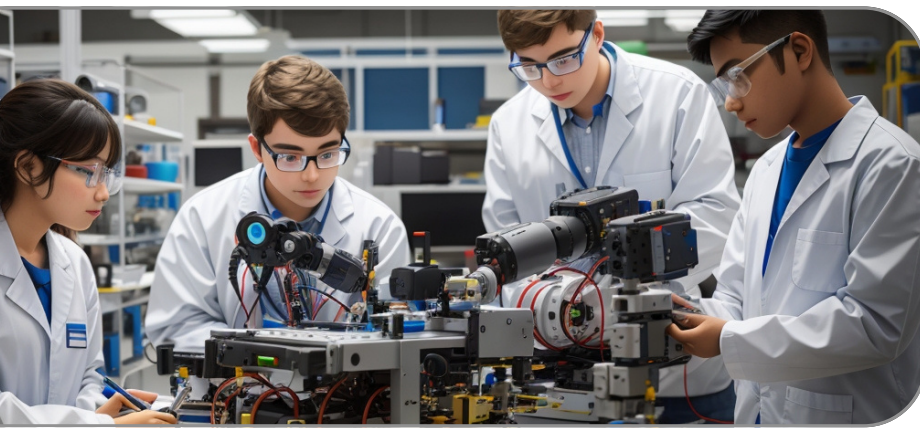
```
volatile unsigned long tiempoPulso;
volatile unsigned long ultimoPulso;
volatile float velocidadAngular = 0.0;
volatile int contadorVueltas = 0;
const int vueltasObjetivo = 5; // Número deseado de vueltas
```

```
LiquidCrystal_I2C lcd(0x3f, 16, 2); // Dirección del I2C y tamaño de la pantalla
```

```
SoftwareSerial bluetooth(10, 11); // RX, TX para el módulo Bluetooth
```

```
char comando = ' '; // Variable para almacenar el comando recibido
unsigned long tiempoInicio = 0; // Variable para almacenar el tiempo inicial
const unsigned long duracionMaxima = 8000; // Duración máxima del movimiento (8 segundos)
```

```
void setup() {
  pinMode(sensorPin, INPUT);
  pinMode(pin3_motor, OUTPUT);
  pinMode(pin4_motor, OUTPUT);
  pinMode(ENA, OUTPUT);
}
```



Código

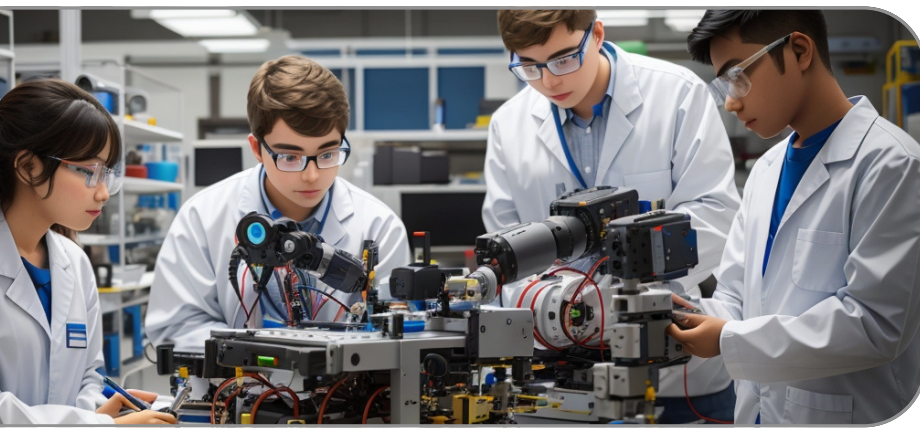
M.C.U.

```
lcd.init();  
lcd.backlight();  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Velocidad: ");  
lcd.setCursor(0, 1);  
lcd.print("RPM");
```

```
bluetooth.begin(9600); // Iniciar comunicación serial con el módulo Bluetooth  
attachInterrupt(digitalPinToInterrupt(sensorPin), detectarPulso, RISING);  
}
```

```
void loop() {  
  mostrarVelocidadEnLCD();  
  controlarMotor();  
  recibirComandoBluetooth();  
}
```

```
void detectarPulso() {  
  tiempoPulso = micros();  
  unsigned long periodo = tiempoPulso - ultimoPulso;  
  ultimoPulso = tiempoPulso;  
  
  if (periodo > 1000) {  
    velocidadAngular = 60000000.0 / periodo;  
    contadorVueltas++;  
  }  
}
```

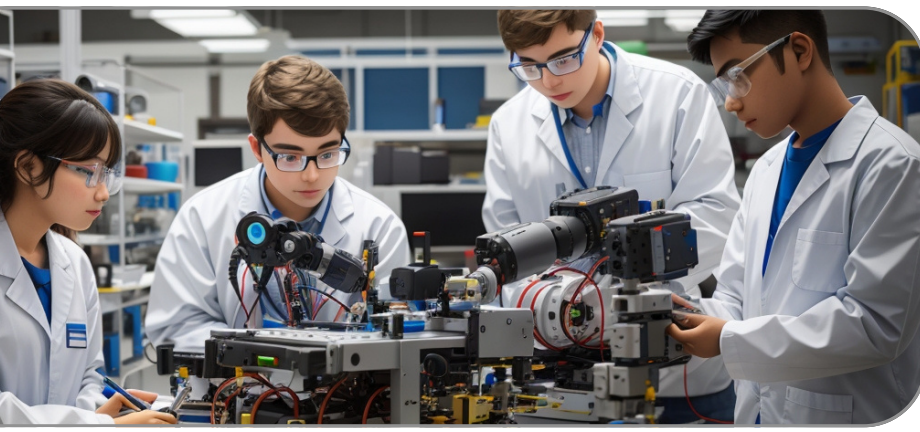


Código

M.C.U.

```
void mostrarVelocidadEnLCD() {  
  lcd.setCursor(10, 0);  
  lcd.print("  ");  
  lcd.setCursor(10, 0);  
  lcd.print(velocidadAngular, 1);  
}
```

```
void controlarMotor() {  
  // Control del motor basado en los comandos recibidos por Bluetooth y el  
  temporizador  
  if (millis() - tiempoInicio < duracionMaxima) {  
    if (comando == 'A') {  
      // Mantener la velocidad actual  
      // No se hace nada aquí, el motor continúa girando a la velocidad actual  
    } else if (comando == 'B') {  
      // Reducir la velocidad  
      analogWrite(ENA, 50); // Cambiar el valor según la velocidad deseada  
    } else if (comando == 'C') {  
      // Acelerar el motor  
      analogWrite(ENA, 150); // Cambiar el valor según la velocidad deseada  
    } else if (comando == 'D') {  
      // Velocidad intermedia  
      analogWrite(ENA, 100); // Cambiar el valor según la velocidad deseada  
    }  
  }
```



Código

M.C.U.

```
} else {  
  // Si ha pasado el tiempo límite, detener el motor  
  digitalWrite(pin3_motor, LOW);  
  digitalWrite(pin4_motor, LOW);  
  analogWrite(ENA, 0);  
}  
  
void recibirComandoBluetooth() {  
  if (bluetooth.available()) {  
    comando = bluetooth.read(); // Leer el comando enviado por Bluetooth  
    tiempoInicio = millis(); // Actualizar el tiempo inicial al recibir un nuevo comando  
  }  
}
```