

*Código*  
M.R.U.

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

```
SoftwareSerial bluetooth(0, 1);
```

```
int distancia = 0; // Variable para llevar un registro de la distancia recorrida  
int circunferenciaRueda = 20.3; // Circunferencia de la rueda en centímetros
```

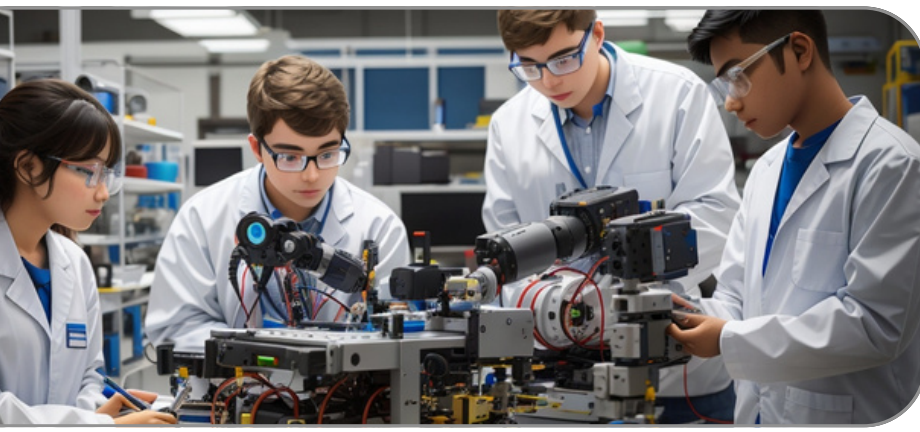
```
const int ri1 = 5;  
const int ri2 = 6;  
const int rd1 = 9;  
const int rd2 = 10;  
int estado;
```

```
void setup() {  
  Serial.begin(9600); // Inicializa la comunicación serial para la depuración  
  bluetooth.begin(9600); // Inicializa la comunicación serial con el módulo Bluetooth  
  pinMode(ri1, OUTPUT);  
  pinMode(ri2, OUTPUT);  
  pinMode(rd1, OUTPUT);  
  pinMode(rd2, OUTPUT);  
}
```

```
void moverAuto(int distanciaDeseada, int duracionMovimiento) {  
  int numVueltas = distanciaDeseada / circunferenciaRueda;
```

```
  for (int i = 0; i < numVueltas; i++) {  
    digitalWrite(ri1, HIGH);  
    digitalWrite(ri2, LOW);  
    digitalWrite(rd1, HIGH);  
    digitalWrite(rd2, LOW);
```

```
    delay(duracionMovimiento); // Ajusta el tiempo de acuerdo a la velocidad
```



# Código

## M.R.U.

```
digitalWrite(ri1, LOW);  
digitalWrite(ri2, LOW);  
digitalWrite(rd1, LOW);  
digitalWrite(rd2, LOW);
```

```
distancia += circunferenciaRueda;  
}  
}
```

```
void loop() {  
  if (bluetooth.available() > 0) {  
    char command = bluetooth.read();  
  
    switch (command) {  
      case '1':  
        moverAuto(20, 1000); // Mueve el auto 20 cm a velocidad normal  
        break;  
      case '2':  
        moverAuto(30, 2000); // Mueve el auto 30 cm más lento (duración más larga)  
        break;  
      case '3':  
        moverAuto(35, 1000); // Mueve el auto 35 cm a velocidad normal  
        break;  
      case '4':  
        moverAuto(40, 800); // Mueve el auto 40 cm más rápido (duración más corta)  
        break;  
    }  
  }  
}
```