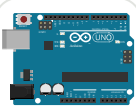


Sorting machine: teach it to see · START A BIT

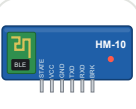


Scan to open

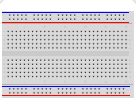
Parts you need



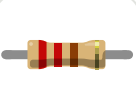
Arduino
UNO



Bluetooth
module
HM-10



Breadboard



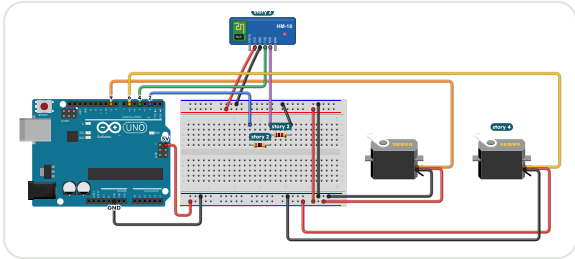
1 kΩ and 2
kΩ resistors



FS90 servo
(180°): the
tray



A second
FS90 servo:
the gate



- pin 9 → tray servo (orange wire)
- 5V → + rail → servos and the HM-10
- GND → – rail → servos and the HM-10
- pin 4 ← HM-10 TXD
- pin 2 → 1 kΩ → HM-10 RXD; 2 kΩ → – rail
- pin 6 → gate servo (orange wire)

story
2

story
2

story
4

The resistors: 1 kΩ is brown, black, red; 2 kΩ is red, black, red. If the Arduino restarts when a servo moves, give the servos their own battery pack (GND joined).

Story 1: Tip the tray

```
// Story 1: Tip the tray. Type left or right in the terminal.
#include <Servo.h>

const int trayPin = 9;
const int level = 90;
const int left = 45;
const int right = 135;
Servo tray;

void setup() {
  Serial.begin(9600);
  tray.attach(trayPin);
  tray.write(level);
  Serial.println("Type left or right");
}

void tip(int angle) {
  tray.write(angle);
  delay(600);
  tray.write(level);
}

void loop() {
  if (Serial.available()) {
    String word = Serial.readStringUntil('\n');
    word.trim();
    if (word == "left") {
      tip(left);
    } else if (word == "right") {
      tip(right);
    }
  }
}
```

Story 2: Show what the AI sees

```
// Story 2: Show what the AI sees. The AI link sends a command to the Arduino.
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(4, 2); // pin 4 hears the AI, pin 2 sends to the AI

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  if (bluetooth.available()) {
    String word = bluetooth.readStringUntil('\n');
    word.trim();
    Serial.println(word);
    if (word == "red") {
      digitalWrite(LED_BUILTIN, HIGH);
    } else {
      digitalWrite(LED_BUILTIN, LOW);
    }
  }
}
```

Then teach it your beans

1. Open [Teachable Machine](#), make three classes: **red**, **yellow**, **empty**.
2. Point the camera at the tray. For each class, record about 40 pictures: beans in different places and turned, and the empty tray.
3. **Train Model**, then test it with new beans. Then **Export Model**, **Upload my model**, copy the link.
4. In the AI link paste the link at **1 The model** and press **Open**. Test again: now your model decides.

Story 3: Let it decide

```
// Story 3: Let it decide. Red tips the tray left, yellow tips the tray right.
#include <Servo.h>
#include <SoftwareSerial.h>

const int trayPin = 9;
const int level = 90;
const int left = 45;
const int right = 135;
Servo tray;
SoftwareSerial bluetooth(4, 2); // pin 4 hears the AI, pin 2 sends to the AI

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
  tray.attach(trayPin);
  tray.write(level);
}

void tip(int angle) {
  tray.write(angle);
  delay(600);
  tray.write(level);
}

void loop() {
  if (bluetooth.available()) {
    String word = bluetooth.readStringUntil('\n');
    word.trim();
    Serial.println(word);
    if (word == "red") {
      tip(left);
    } else if (word == "yellow") {
      tip(right);
    }
  }
}
```

Story 4: One at a time

```
// Story 4: One at a time. The gate lets one bean through. The AI decides: tip left, tip right, or empty.
#include <Servo.h>
#include <SoftwareSerial.h>

const int trayPin = 9;
const int gatePin = 6;
const int level = 90;
const int left = 45;
const int right = 135;
const int hold = 60;
const int open = 120;
Servo tray;
Servo gate;
SoftwareSerial bluetooth(4, 2); // pin 4 hears the AI, pin 2 sends to the AI

int sorted = 0;
int tries = 0;

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
  tray.attach(trayPin);
  gate.attach(gatePin);
  tray.write(level);
  gate.write(hold);
  Serial.println("Type go to start");
}

void letOneIn() {
  gate.write(open);
  delay(250);
  gate.write(hold);
  delay(700); // let the bean through
  bluetooth.println("?");
}

void tip(int angle) {
  tray.write(angle);
  delay(600);
  tray.write(level);
  delay(400);
  sorted = sorted + 1;
  Serial.print("sorted: ");
  Serial.println(sorted);
}

void loop() {
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "go") {
      tries = 0;
      letOneIn();
    }
  }
  if (bluetooth.available()) {
    String word = bluetooth.readStringUntil('\n');
    word.trim();
    if (word == "red") {
      tries = 0;
      tip(left);
      letOneIn();
    } else if (word == "yellow") {
      tries = 0;
      tip(right);
      letOneIn();
    } else if (word == "empty") {
      tries = tries + 1;
      if (tries < 3) {
        letOneIn();
      } else {
        Serial.println("Empty or stuck: check the beans");
      }
    } else {
      delay(300); // "unsure": 300ms
      bluetooth.println("?");
    }
  }
}
```