

The Remote Control Cardboard Box

March 2019

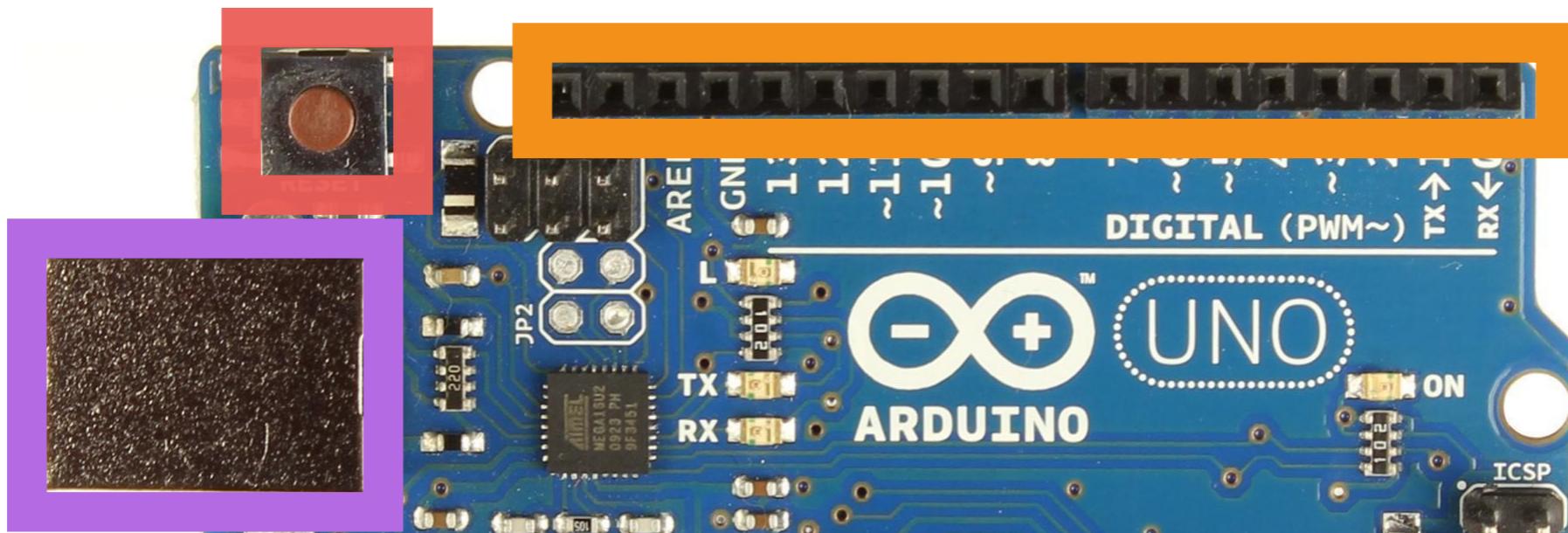
Arduino & Code

Remote Control

Motor Control

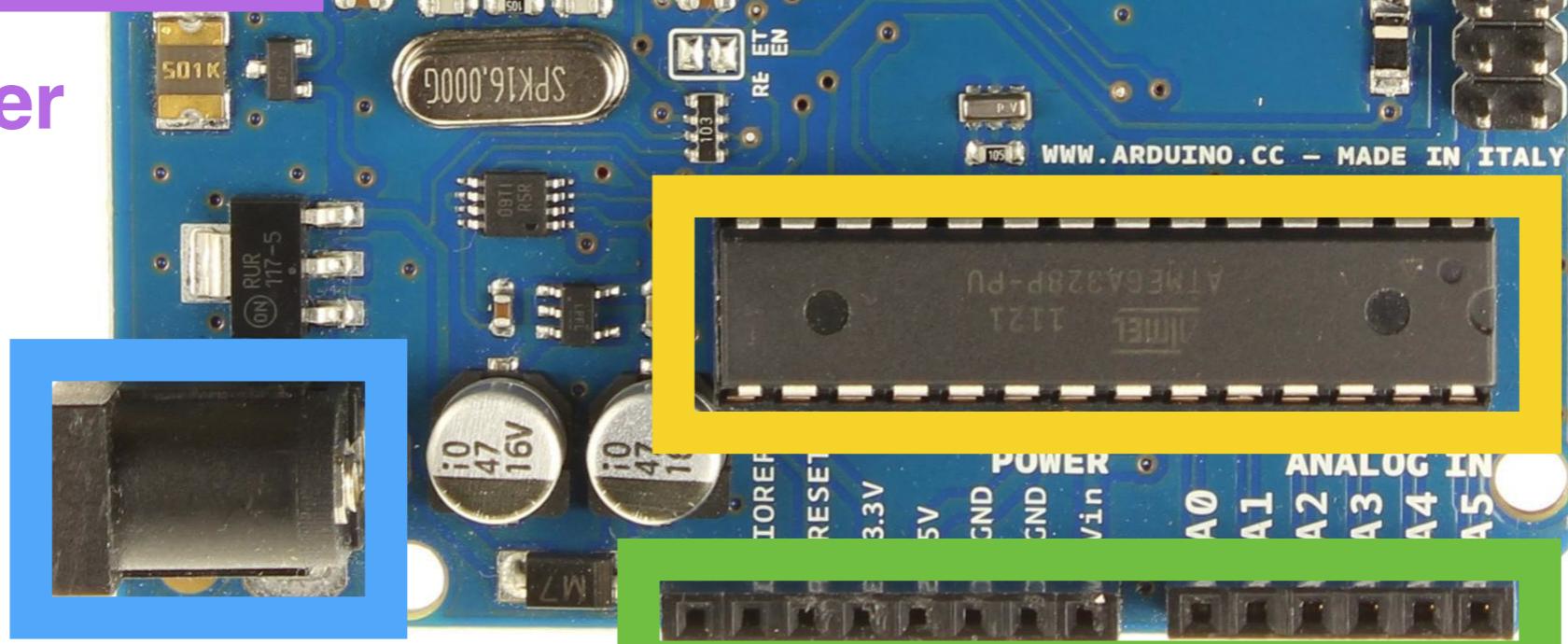
Merging & Integration

Reset Button



Digital Inputs & Outputs

USB / Power



ATmega 328

DC Power

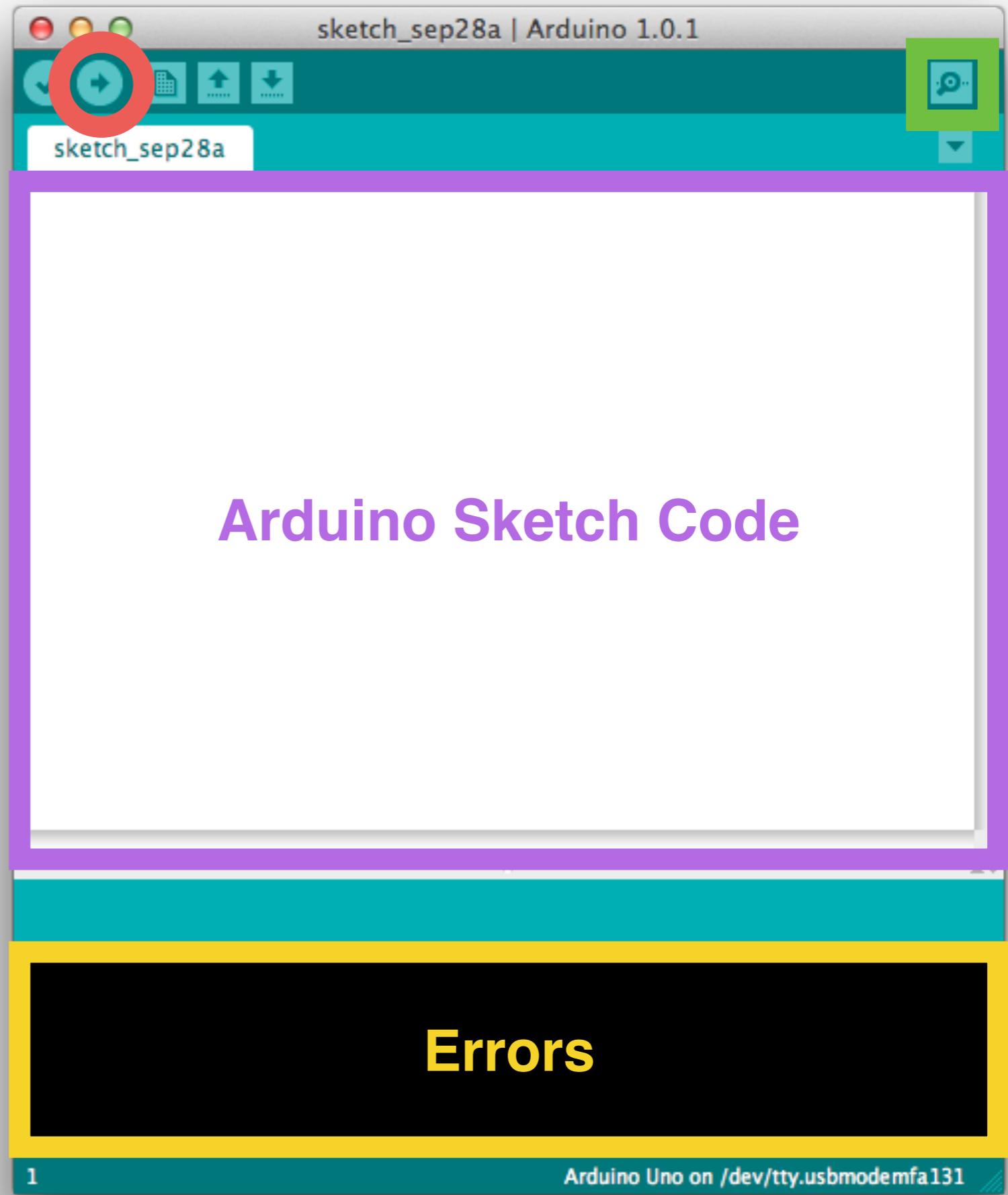
Power & Analog Inputs

Arduino Hardware: UNO

Arduino Software

Upload

Serial Monitor



Anatomy of a Sketch

Start Comment

Blink

Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

End Comment

Descriptive Comment

Start Comment

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}
```

Line Comments

```
// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);           // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000);           // wait for a second
}
```

File > Examples > Basics > Blink

Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/



// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);           // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000);           // wait for a second
}
```

File > Examples > Basics > Blink

Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

*/
Hey Arduino,
here's how you setup

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000);
}
```

File > Examples > Basics > Blink

Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/



// the setup function runs once when you press reset or power the board
void setup() {
    // initializes the digital pin as an output.
    pin
}

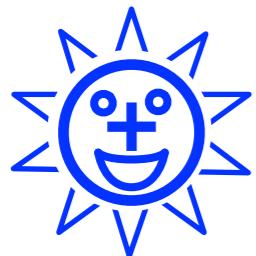
// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                    // wait for a second
    digitalWrite(13, LOW);          // make the voltage LOW
    delay(1000);                    // wait for a second
}
```

Hey Arduino,
here's how you **loop**

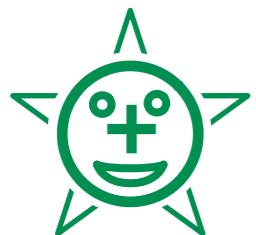
Code block for **loop**

Electronics is...

Moving Charge



Highly energetic charge particle

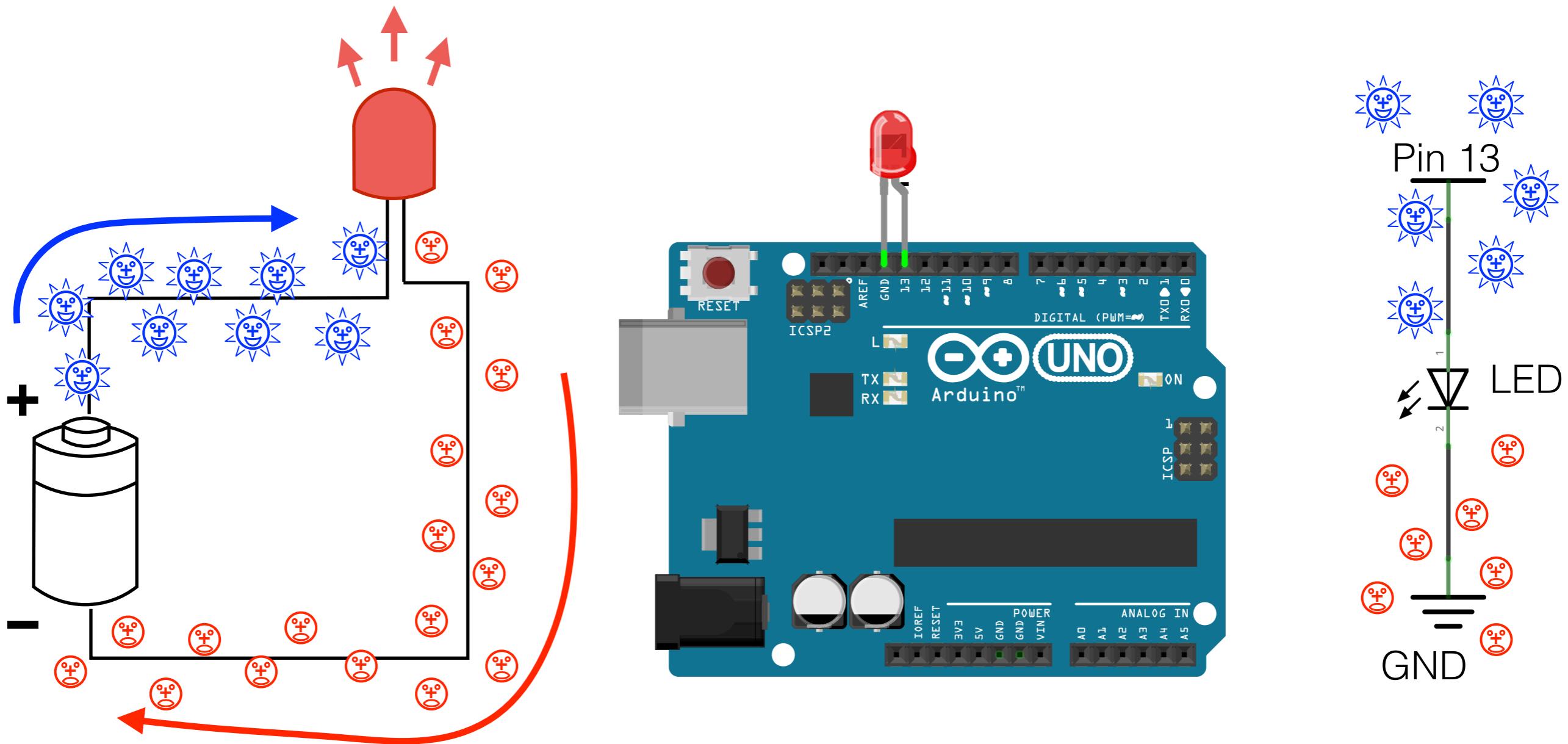


Less energetic charge particle



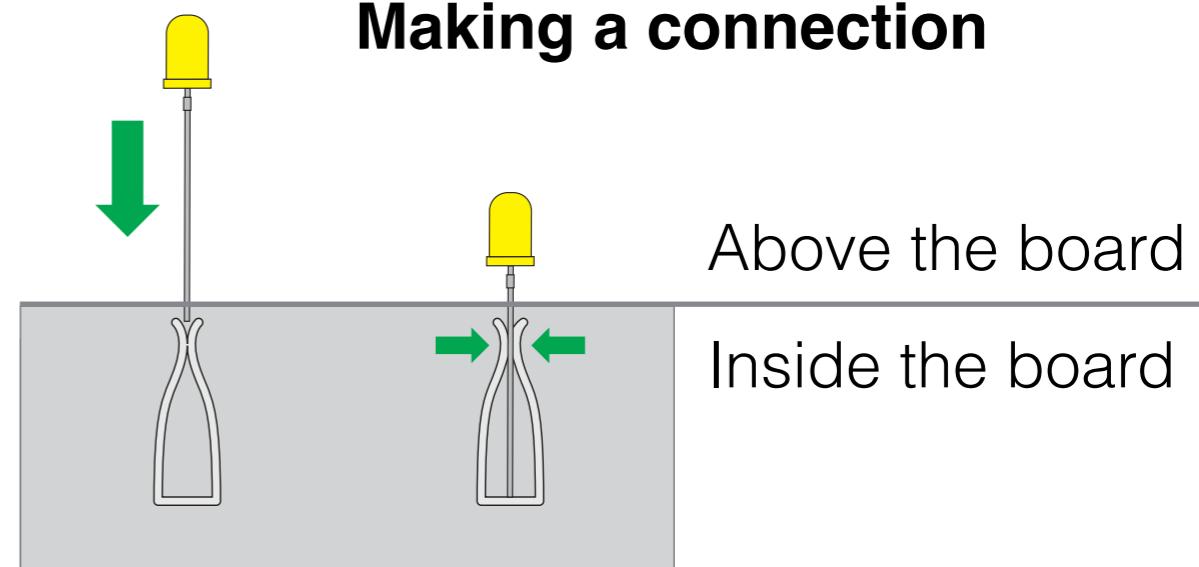
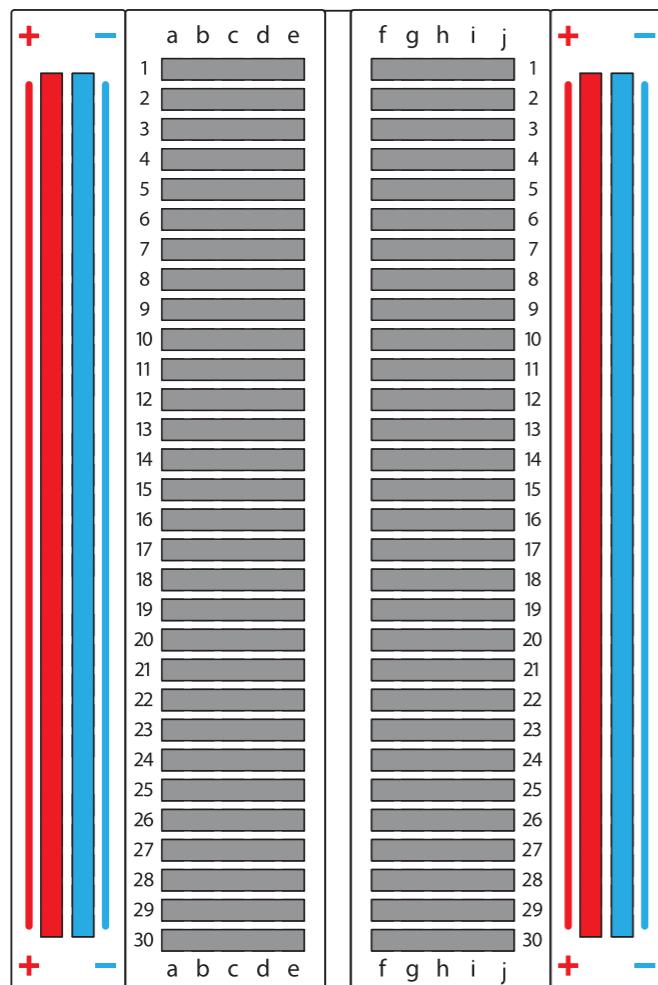
Exhausted charge particle

Simple Circuit



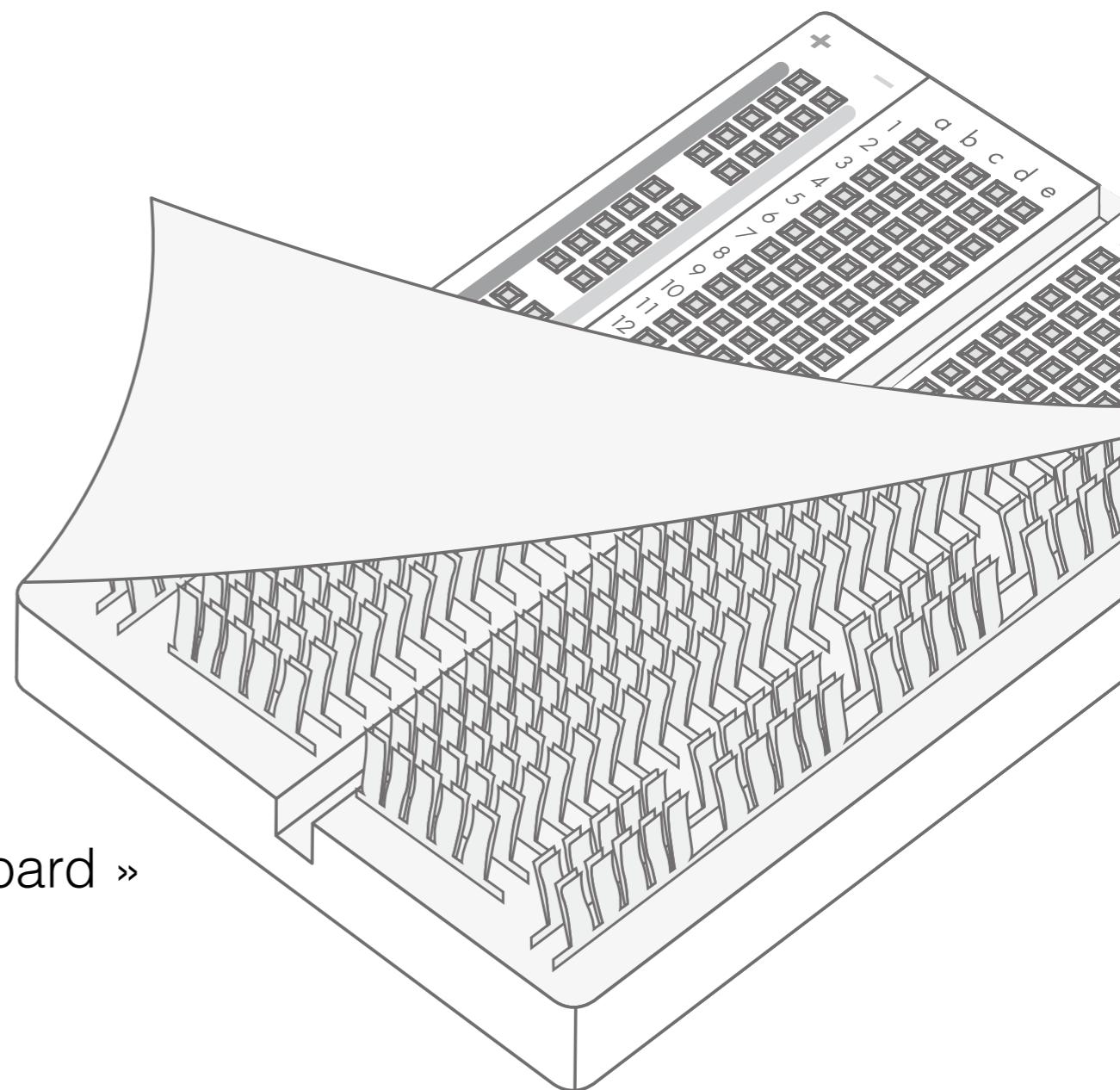
The Breadboard

- + Runs power along column
- Runs ground along column
- Each numbered row has 5 connected sockets



Above the board

Inside the board



Inside the board »

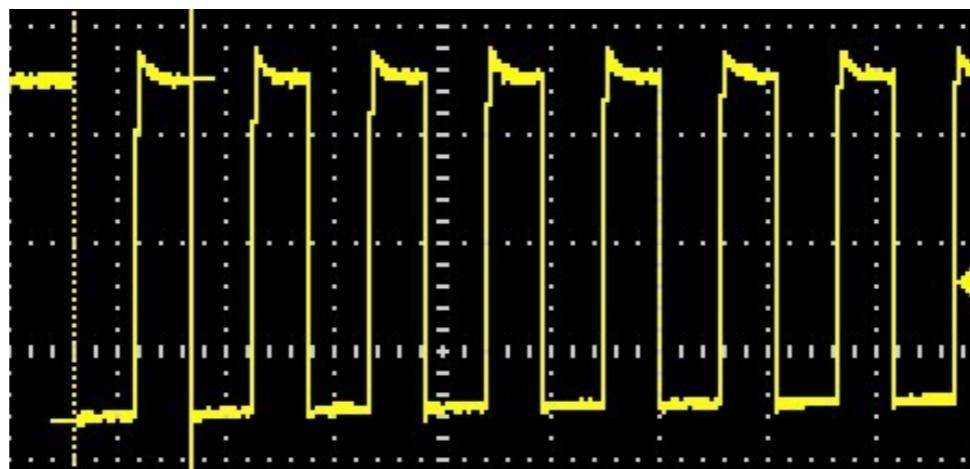
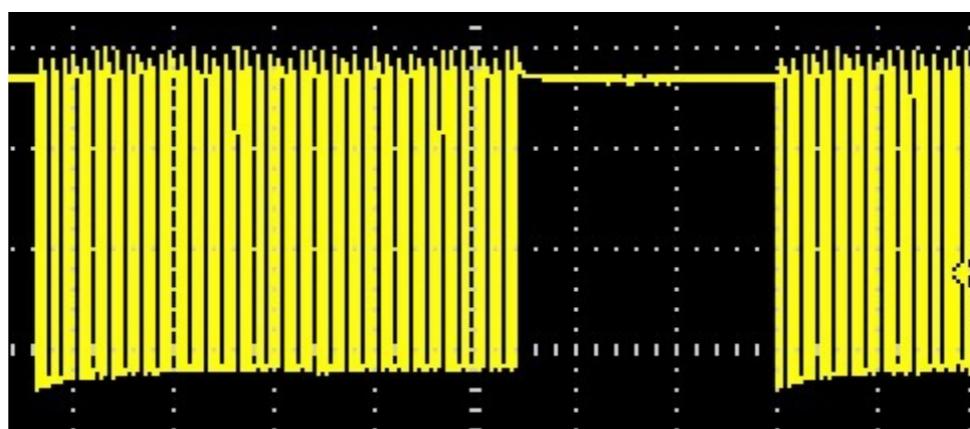
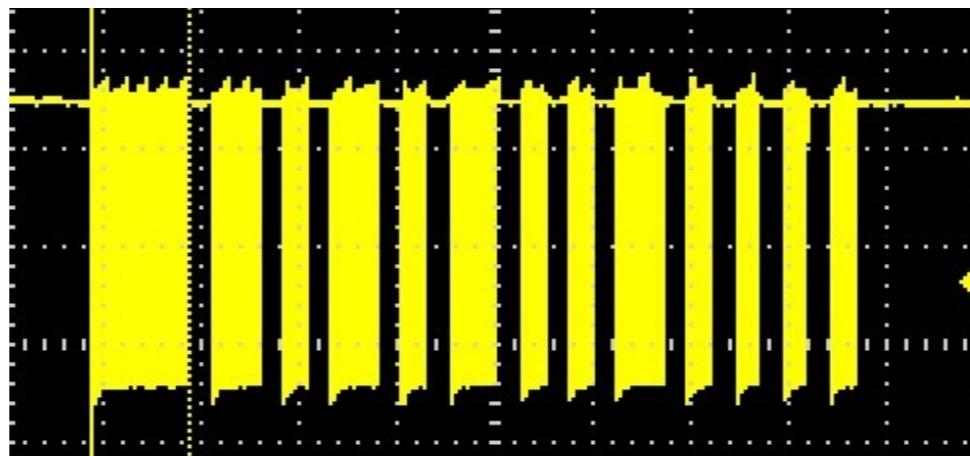
Arduino & Code

Remote Control

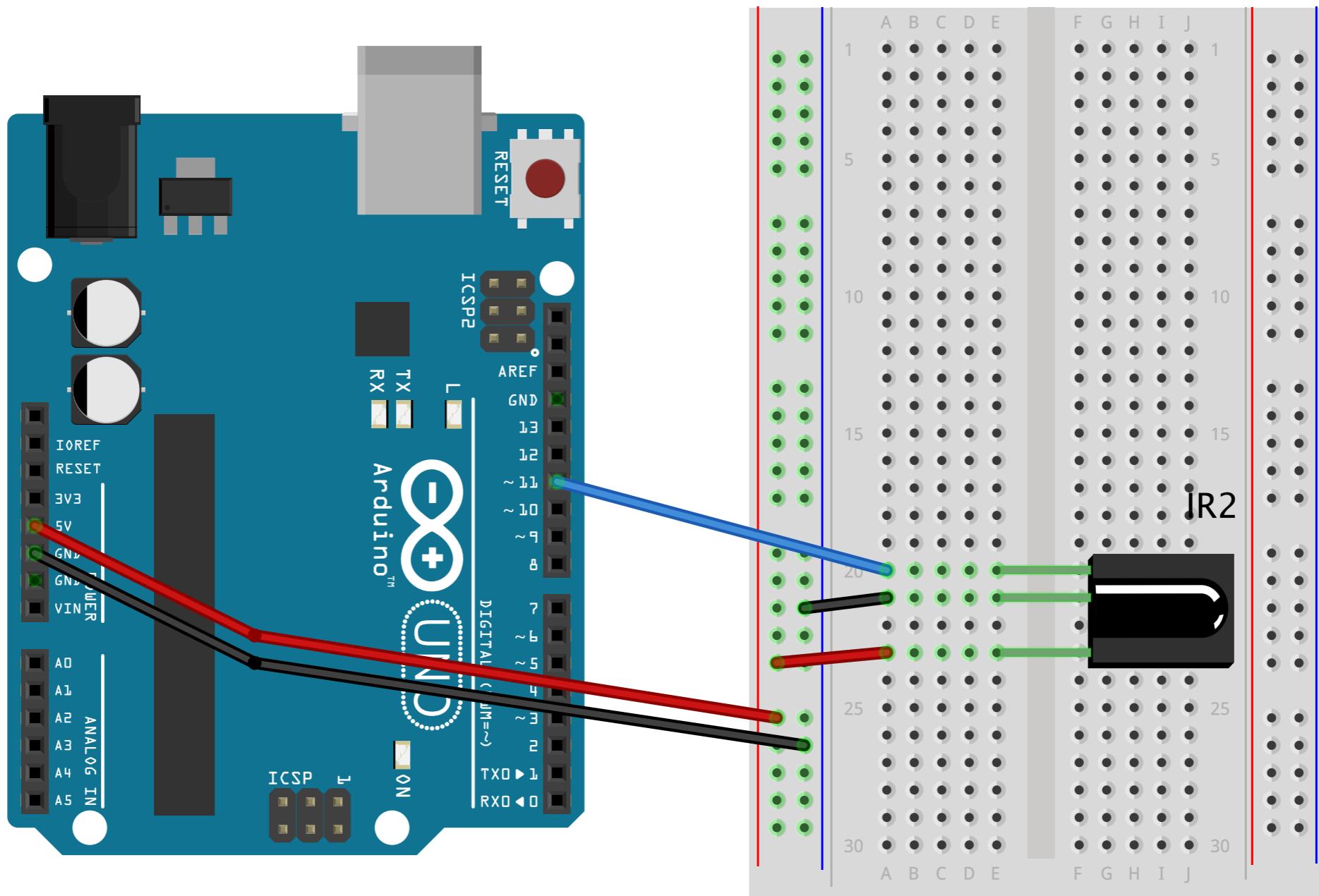
Motor Control

Merging & Integration

IR Theory



IR Receiver



IR Receiver Library

- We'll need to use the IRremote library to use this IR Receiver.
- Visit Sketch > Include Library > Manage Libraries...
- Filter search for IRremote
- Click the Install button

IR Programming

```
/*
IR Receiver Print
Reads values from the IR receiver and print them over Serial.
*/

#include <IRremote.h>

int remoteInputPin = 2;
IRrecv receiver(remoteInputPin);

decode_results results;

void setup()
{
    Serial.begin(9600);
    receiver.enableIRIn();
}

void loop() {
    if (receiver.decode(&results)) {
        Serial.println(results.value, HEX);

        if (results.value == 0xABCD) {
            Serial.println("FORWARD");
        } else if (results.value == 0x0000) {
            Serial.println("HALT");
        }
    }

    receiver.resume();
}
}
```

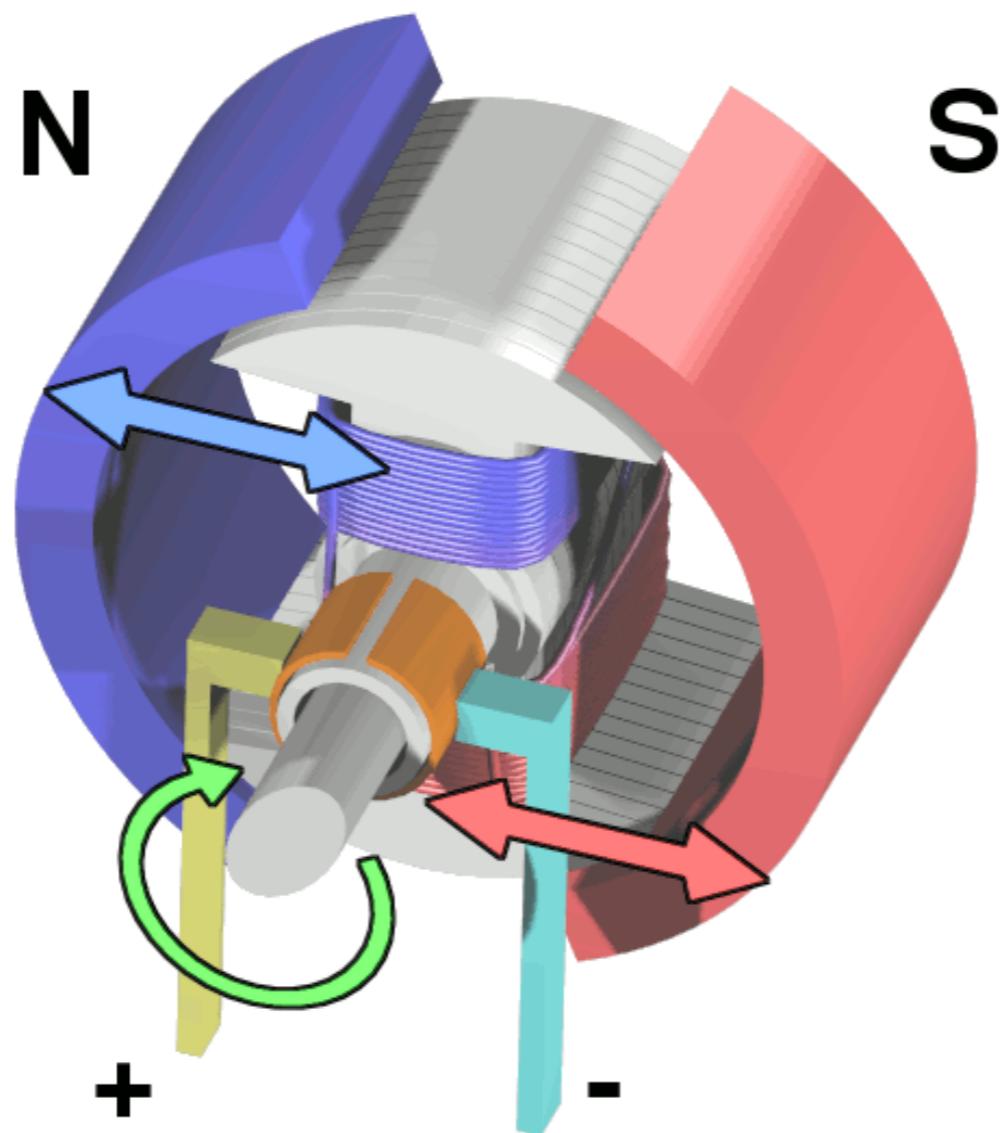
Arduino & Code

Remote Control

Motor Control

Merging & Integration

Electric Motor

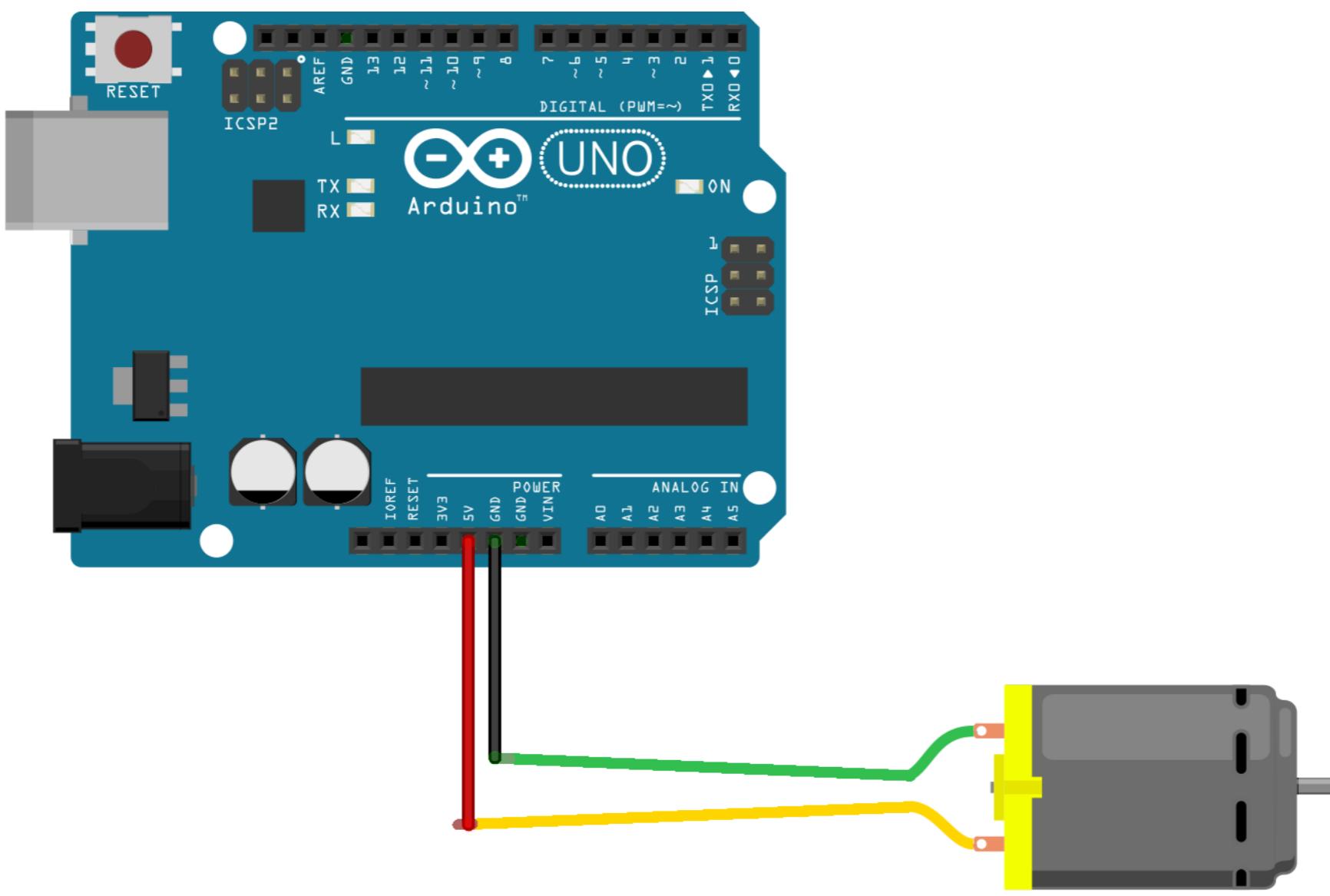


[wikipedia]

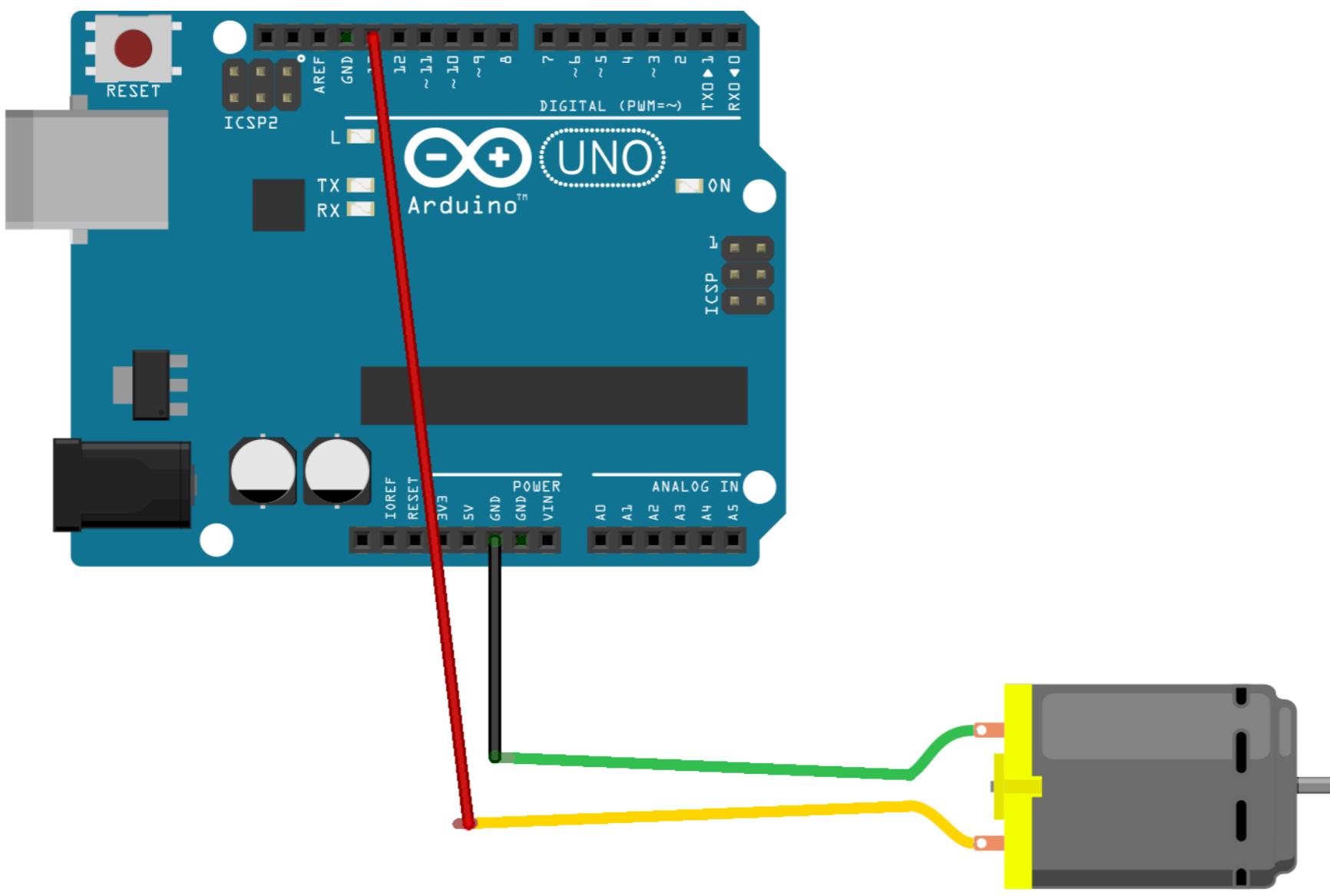
The Gearmotor



Testing the Motor



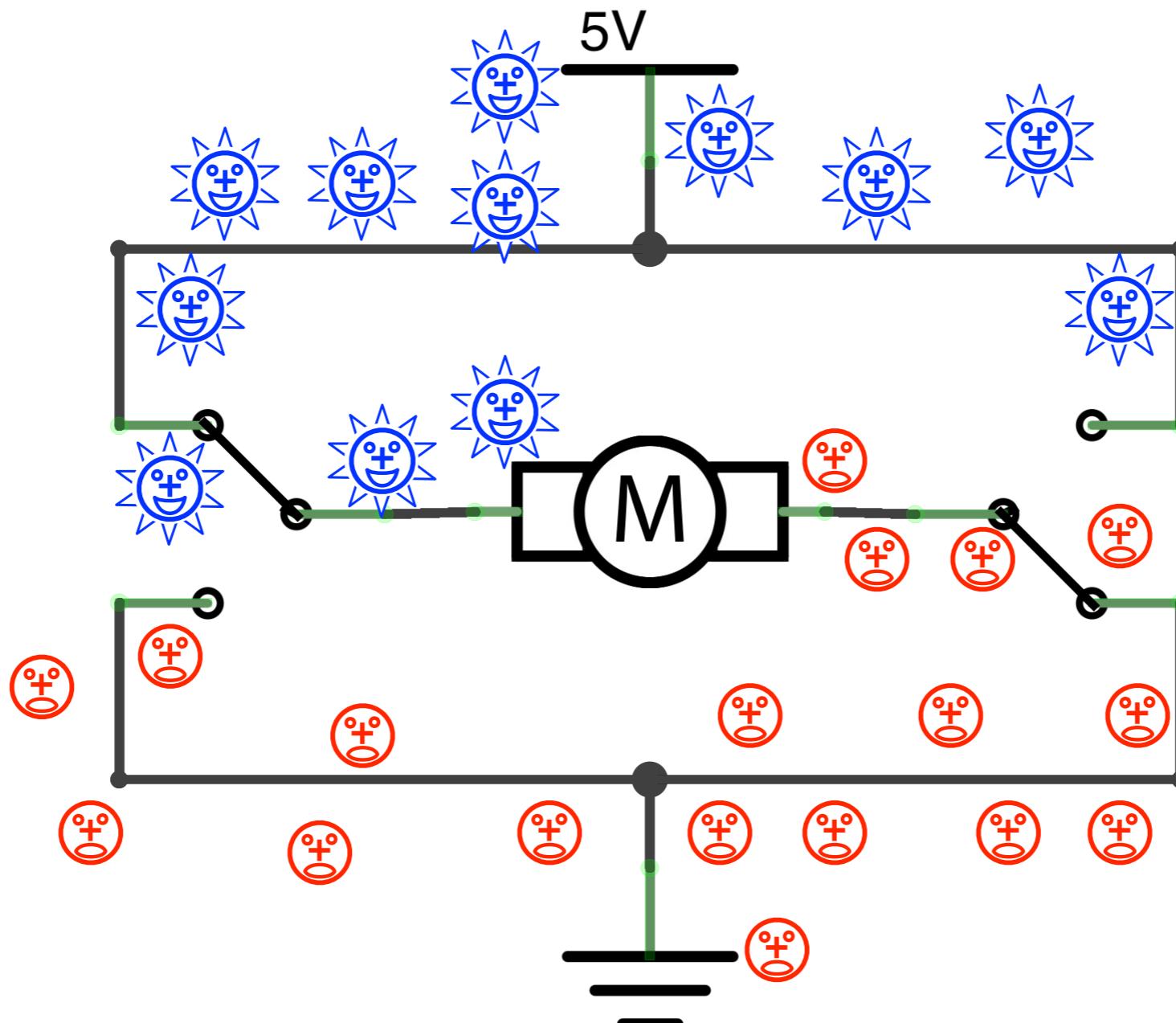
Testing the Motor



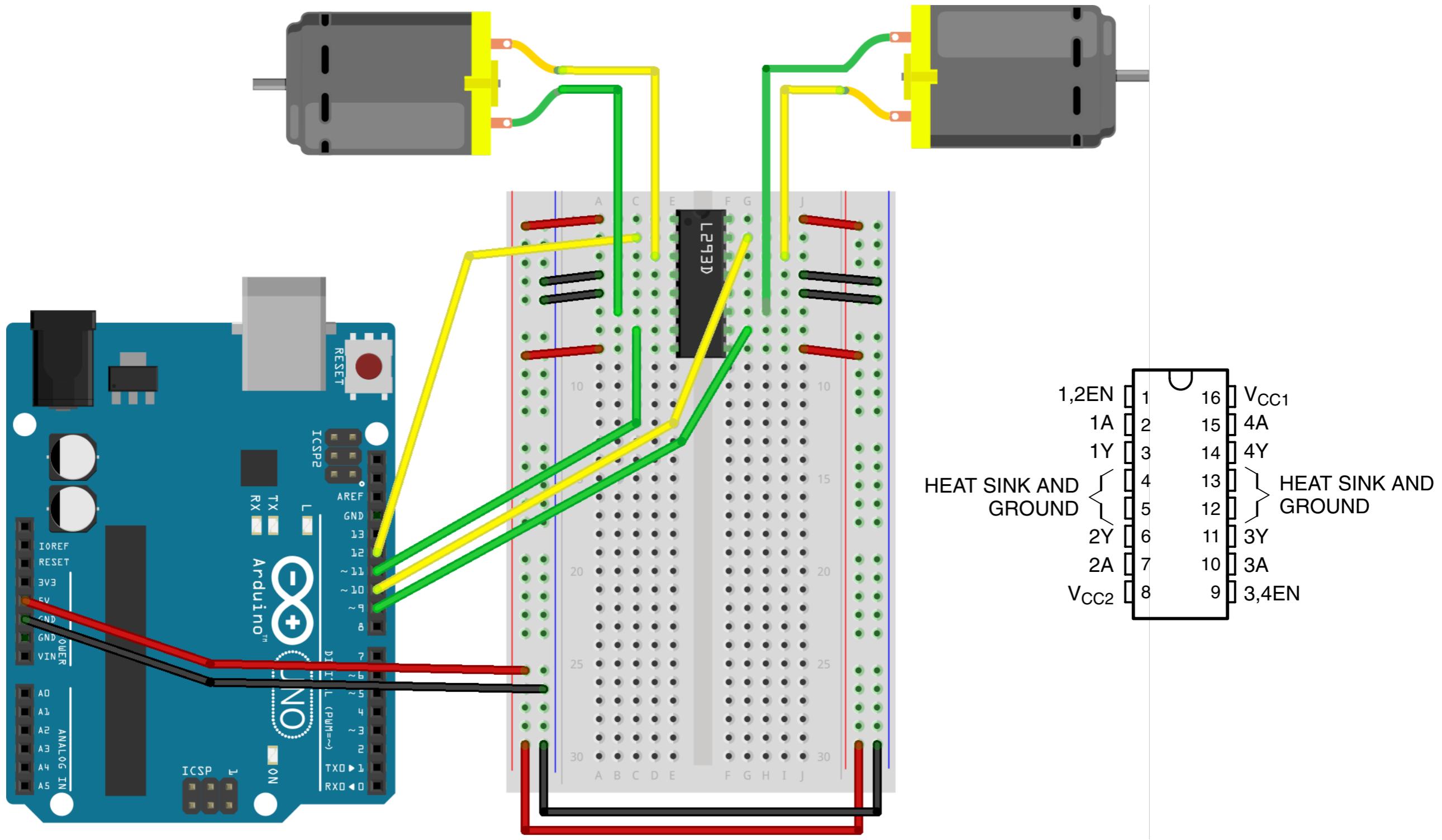


problem?

Pin-controlled Switch!



Motor Driver



Motor Programming

```
/*
 Motor Tester
 Scaffolding to figure out
 motor directions.
 */

int motorRightForward = 9;
int motorRightReverse = 10;
int motorLeftForward = 11;
int motorLeftReverse = 12;

void setup() {
    pinMode(motorRightForward, OUTPUT);
    pinMode(motorRightReverse, OUTPUT);
    pinMode(motorLeftForward, OUTPUT);
    pinMode(motorLeftReverse, OUTPUT);
}

void forward() {
    digitalWrite(motorRightForward, HIGH);
    digitalWrite(motorRightReverse, LOW);
    digitalWrite(motorLeftForward, HIGH);
    digitalWrite(motorLeftReverse, LOW);
}

void halt() {
    digitalWrite(motorRightForward, LOW);
    digitalWrite(motorRightReverse, LOW);
    digitalWrite(motorLeftForward, LOW);
    digitalWrite(motorLeftReverse, LOW);
}

void loop() {
    forward();
    delay(1000);
    halt();
    delay(1000);
}
```

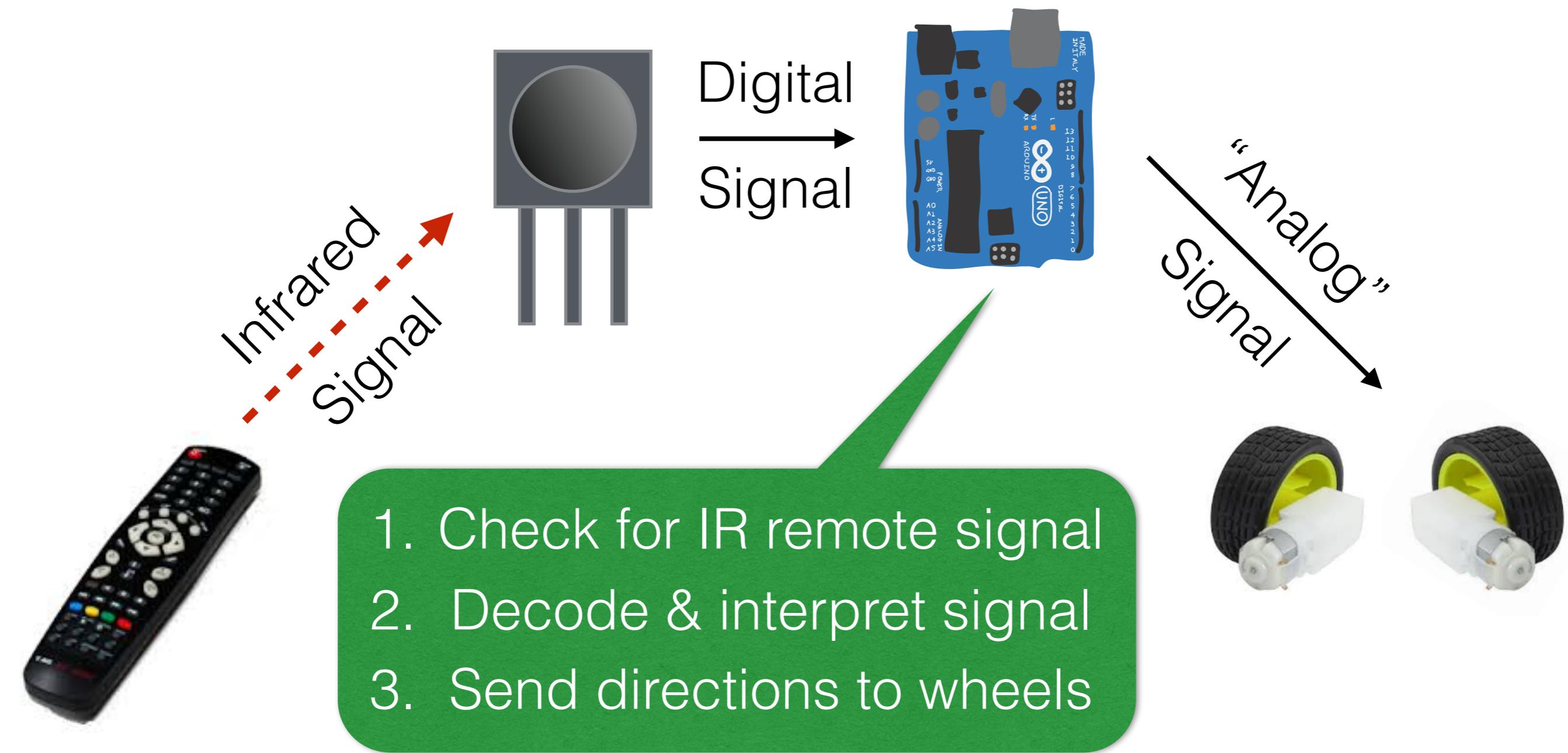
Arduino & Code

Remote Control

Motor Control

Merging & Integration

System Integration



Merging Sketches

1. Use the remote-testing code as the **base** sketch; copy the **motor** sketch into this one.
2. Copy the **motor code** that's **outside of the setup and loop functions** into the **base** sketch: **keep it outside** the setup and loop functions.
3. Copy the **motor code** that's **inside the setup function** into the **base** sketch: **keep it inside** the setup function.
4. Update the **base** sketch; where you currently have `Serial.println("FORWARD")`, etc., add a call to the forward function: `forward();`