Introduction to Embedded Systems (IES)

Module 4
Interfacing to actuators

Version 2023-01-23, Kjeld Jensen kjen@sdu.dk

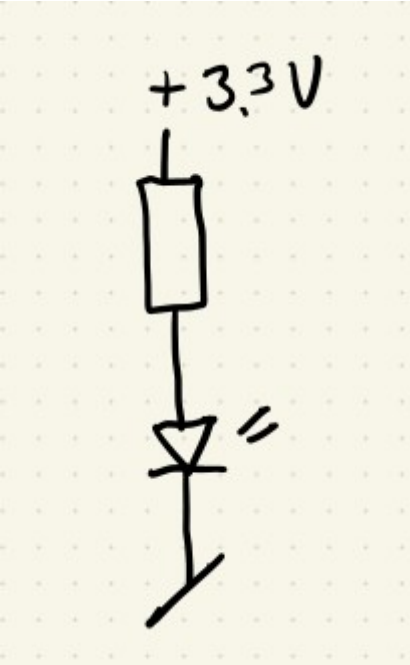
In this module we will interface actuators to the Raspberry Pi Pico microcontroller and explore how to control these using the Arduino programming language. We will use LEDs, a Piezo horn and a servo motor as examples.

Agenda

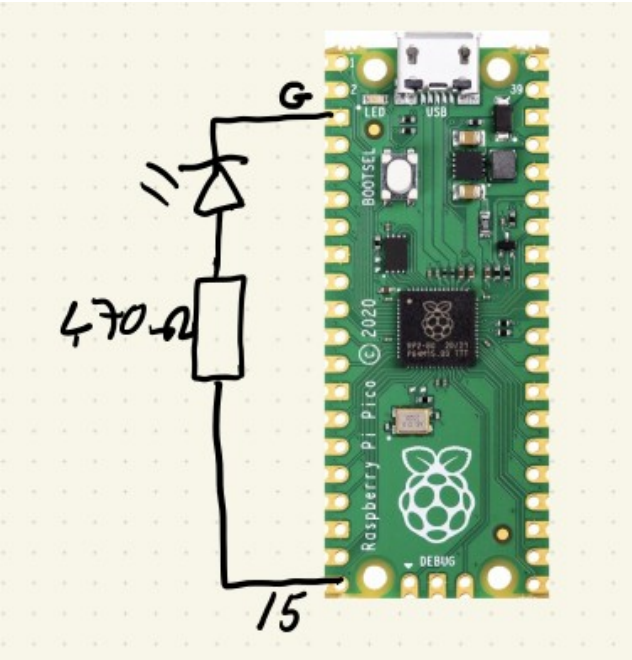
1. Review of module 3
2. Interfacing to actuators exercises
   A) Single LED
   B) Multiple LEDs
   C) RGB LED
   D) Piezo horn
   E) Servo motor

A) Single LED

In this exercise we will connect an external LED to the Pico and control this using the Arduino sketch example `digital_output.ino`



Exercise A.1) Please connect the LED to pin 15 on the Pico using the breadboard using this circuit:



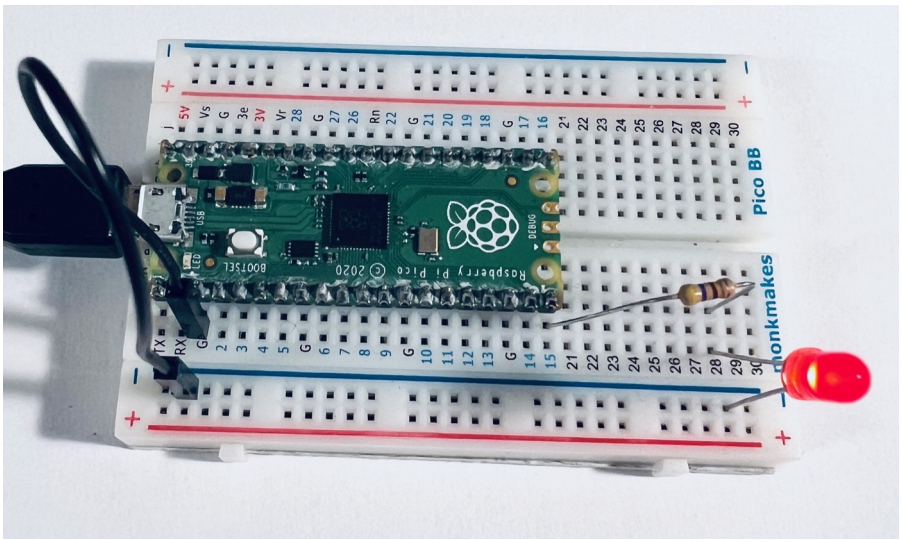
Please remember that the **shortest** pin of the LED is the **negative** pin which must be connected to the Ground **G** pin.

Also please remember to add a 470 Ohm resistor in series with the LED to limit the current.

The 470 Ohm resistor has the color code:

`yellow - violet - brown - gold`

Here is an example of how the breadboard will look:



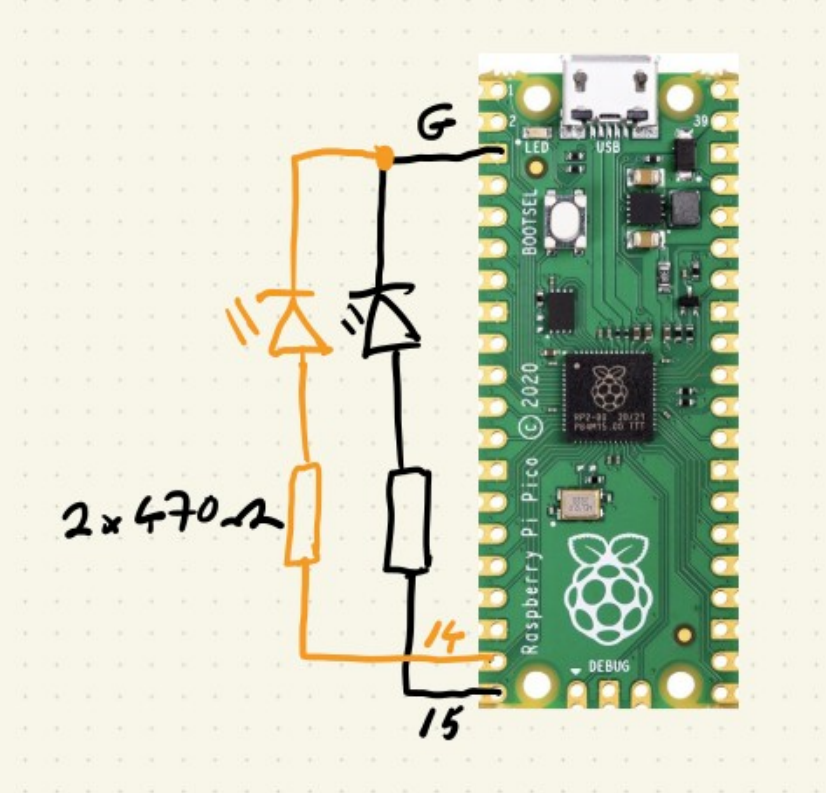
Exercise A.2) Please use the example program below to make the LED blink. A copy is available as `digital_output.ino` in the examples folder.

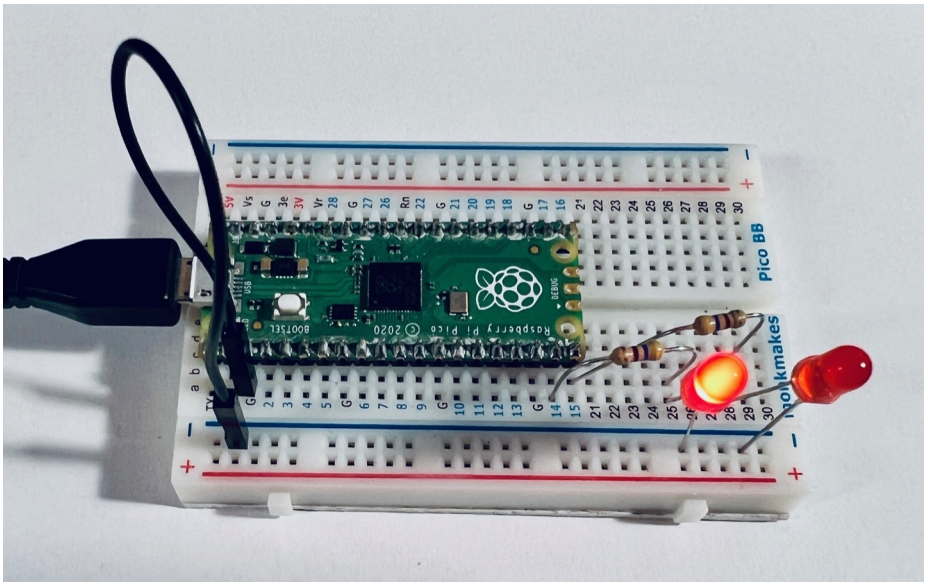
```
#define PIN_DIGI_OUT1 15

void setup() {

  // setup the PIN_DIGI_OUT1 as digital output
  pinMode(PIN_DIGI_OUT1, OUTPUT);
}

void loop() {

  // toggle the PIN_DIGI_OUT1 once
  digitalWrite(PIN_DIGI_OUT1, LOW);
  delay(500);
  digitalWrite(PIN_DIGI_OUT1, HIGH);

  // wait 2 seconds
  delay(2000);
}
```

B) Multiple LEDs

In this exercise we will add a second external LED to the Pico and add control of this to the Arduino sketch `digital_output.ino`

Exercise B.1) Please connect a second LED to pin 14 on the Pico using the breadboard using this circuit:



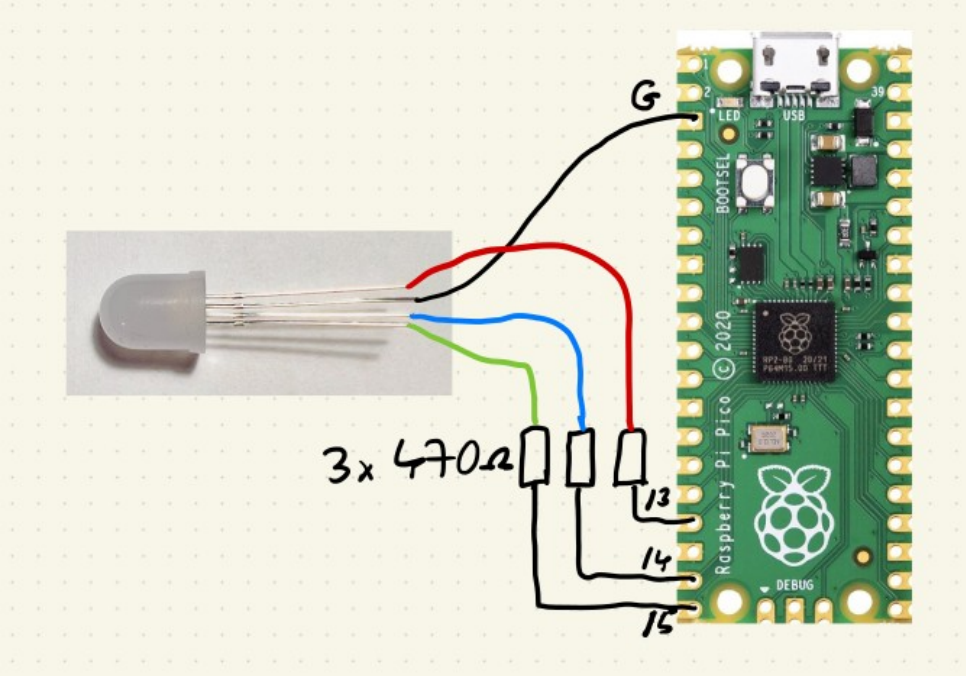


Exercise B.2) Please modify the example program from Exercise A.2 to make both LEDs blink.

C) RGB LED

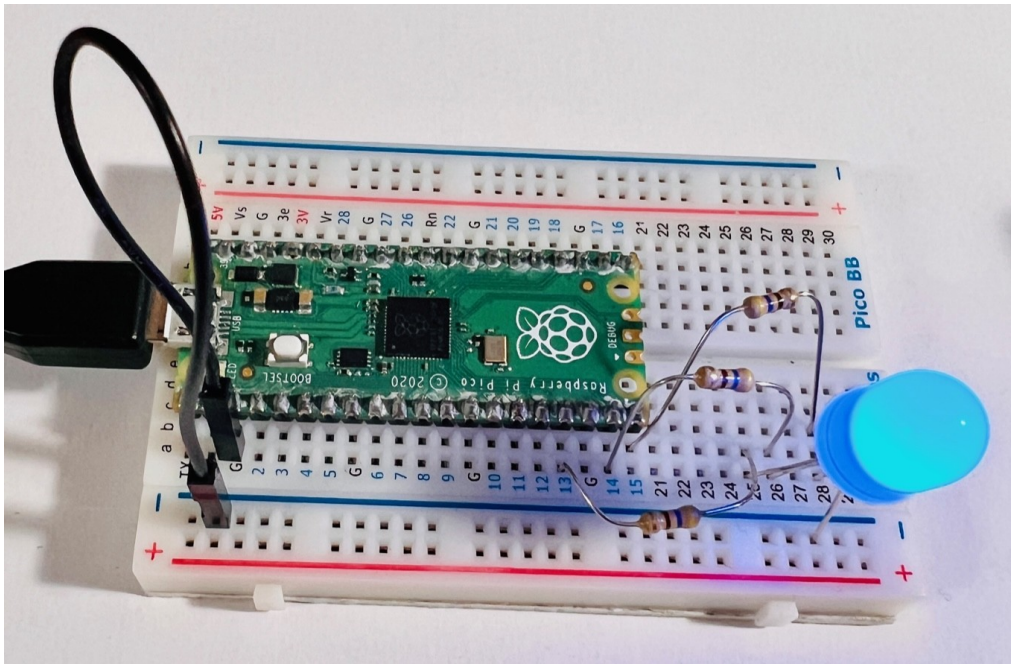
In this exercise we will connect an RGB LED to the Pico and control this using the Arduino sketch example `digital_output.ino` to control this.

Exercise C.1) Please connect the RGB LED to pin 13, 14 and 15 on the Pico using the breadboard using this circuit:



Please remember that the **longest** pin on the RGB LED is the **negative** pin that must be connected to Ground **G**. The three other positive pins produce red, green and blue light.

Also please remember to add a 470 Ohm resistor in series with each of the positive pins of the RGB LED to limit the current.

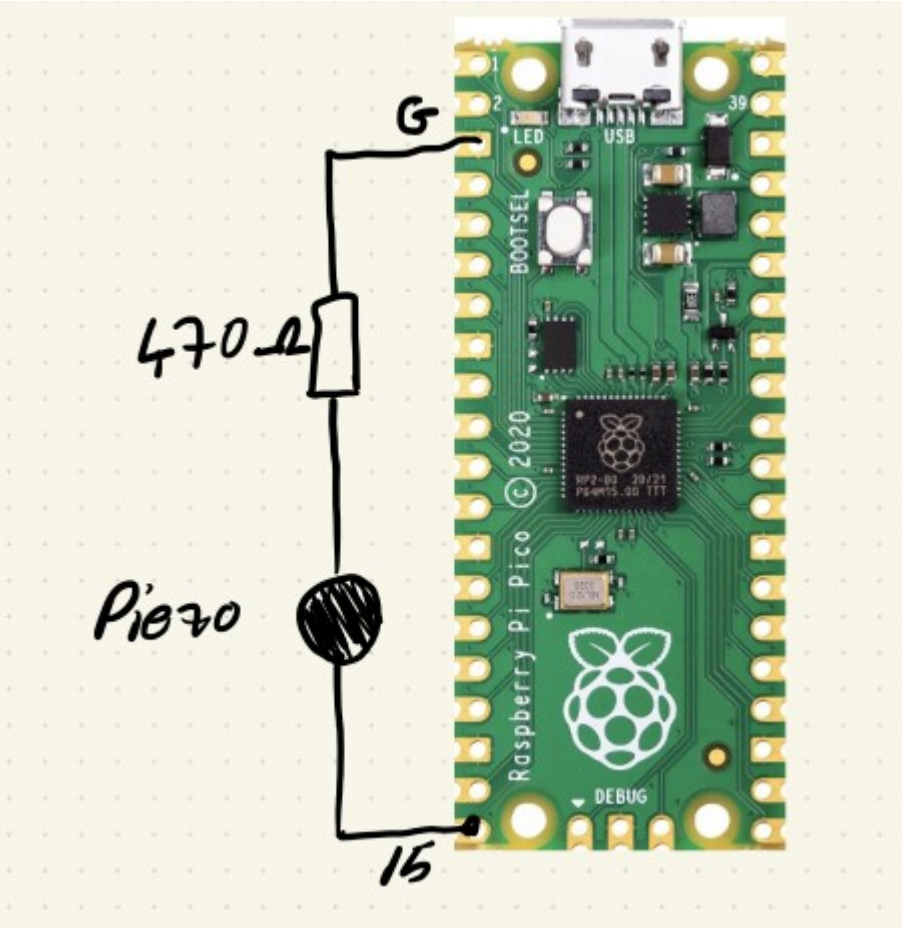


Exercise C.2) Please modify the example program `digital_output.ino` from Exercise A.2 to make all three colors of the RGB LED blink.

D) Piezo horn

In this exercise we will connect a piezo horn to the Pico and control this using the Arduino sketch example `piezo_control.ino`

Exercise D.1) Please connect the piezo horn to pin 15 on the Pico using the breadboard using this circuit:



Please remember to add a 470 Ohm resistor in series with the piezo buzzer to limit the current.

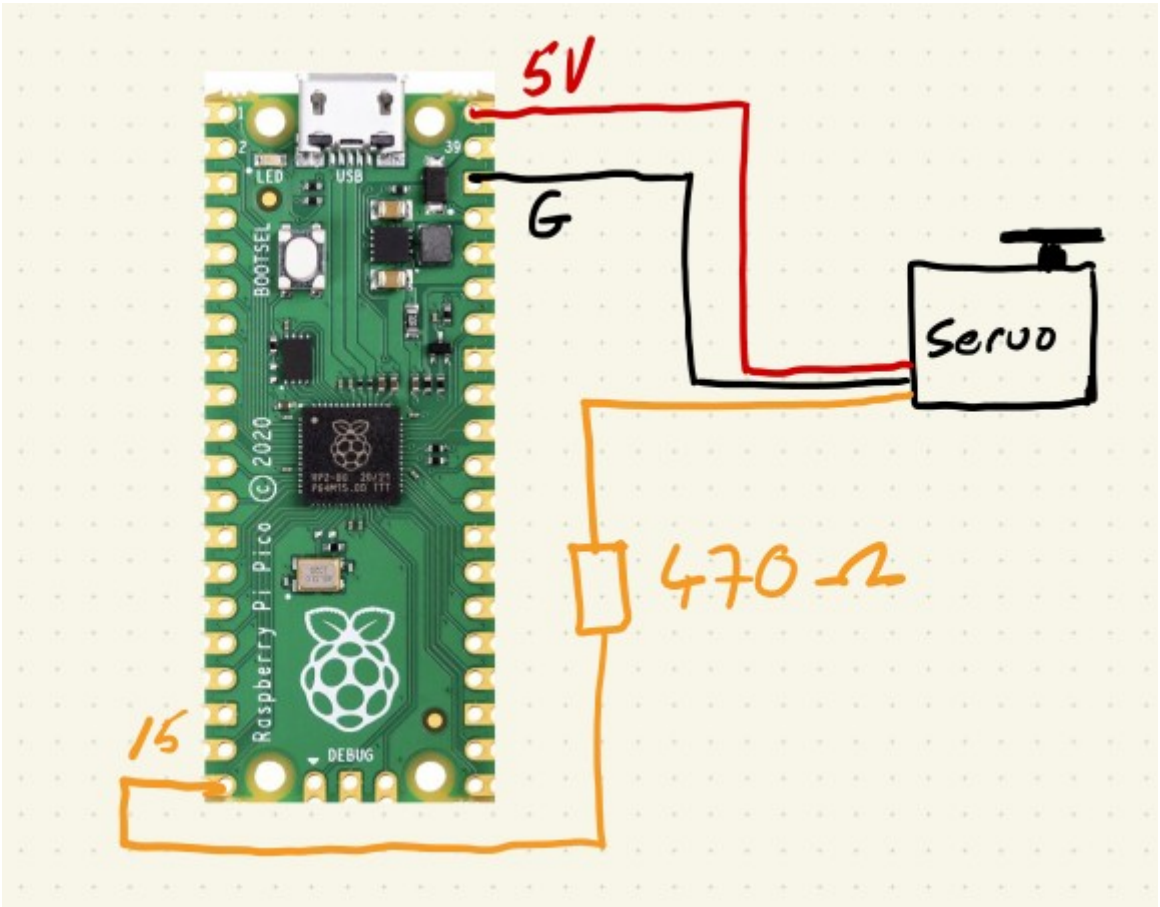
Exercise D.2) Please use the example program below to make the piezo buzzer sound. A copy is available as `piezo_control.ino` in the examples folder.

```
#define PIN_PIEZO 15

void beep(unsigned char ms){

  // turn on the piezo buzzer, almost any value
between 1 and 254 can be used
  analogWrite(PIN_PIEZO, 100);

  // wait ms milliseconds
  delay(ms);

  // turn off the piezo buzzer
  analogWrite (PIN_PIEZO, 0);

  // wait ms milliseconds
  delay(ms);
}

void setup() {

  // setup the PIN_PIEZO as output
  pinMode(PIN_PIEZO, OUTPUT);
}

void loop() {
  beep(200);
  beep(100);
}
```

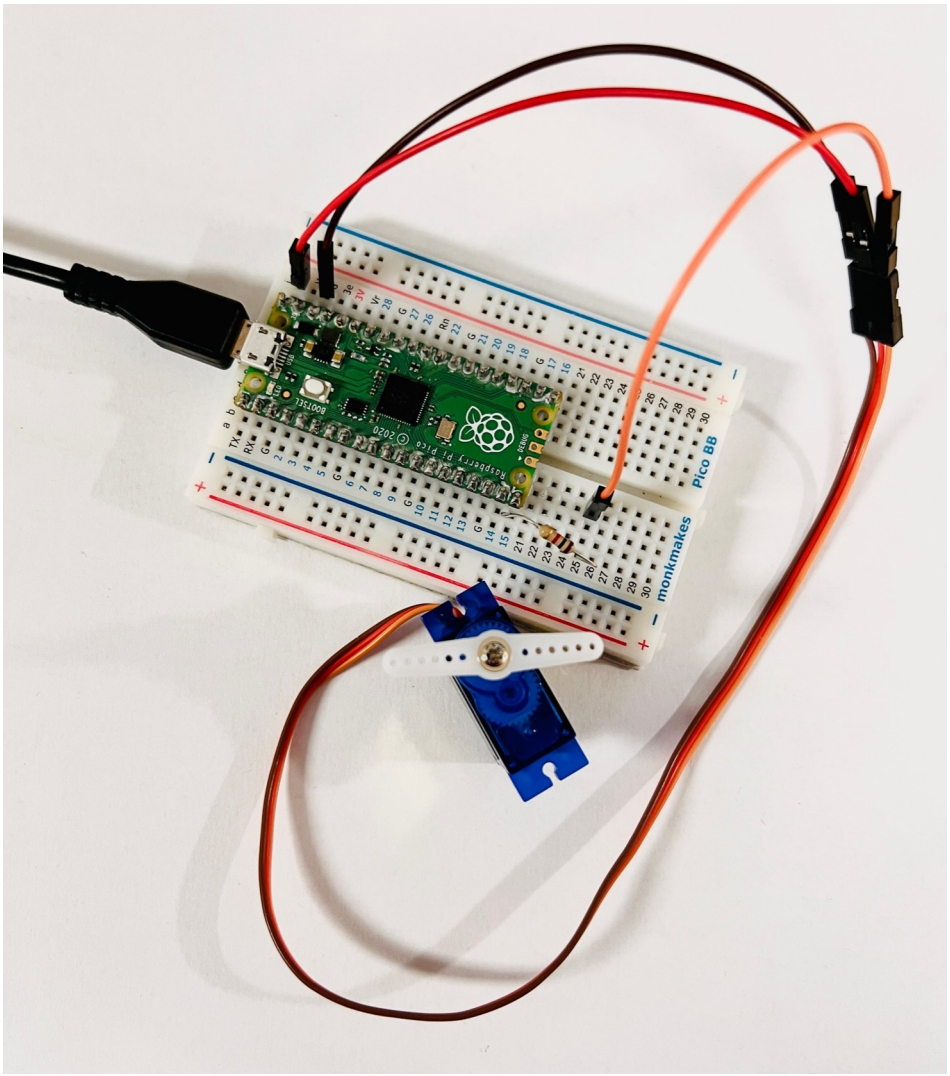
E) Servo motor

In this exercise we will connect a servo motor to the Pico and control this using the Arduino sketch example `servo_control.ino`

Exercise E.1) Please connect the servo motor to pin 15 on the Pico using the breadboard using this circuit:



Please remember to add a 470 Ohm resistor in series with the servo digital input pin to limit the current.



Exercise E.2) Please use the example program below to make the servo move. A copy is available as `servo_control.ino` in the examples folder.

```
#include <Servo.h>

#define PIN_SERVO 15

// create a servo object
Servo myservo;

void setup() {

  // attach PIN_SERVO to the servo object
  myservo.attach(PIN_SERVO);
}

void loop() {
  int i;

  // turn the servo clock wise
  for (i=0; i<=180; i+=1) {
    myservo.write(i);
    delay(10);
  }

  // wait a second
  delay (1000);

  // turn the servo counter clock wise
  for (i=180; i>=0; i-=1) {
    myservo.write(i);
    delay(10);
  }

  // wait a second
  delay (1000);
}
```