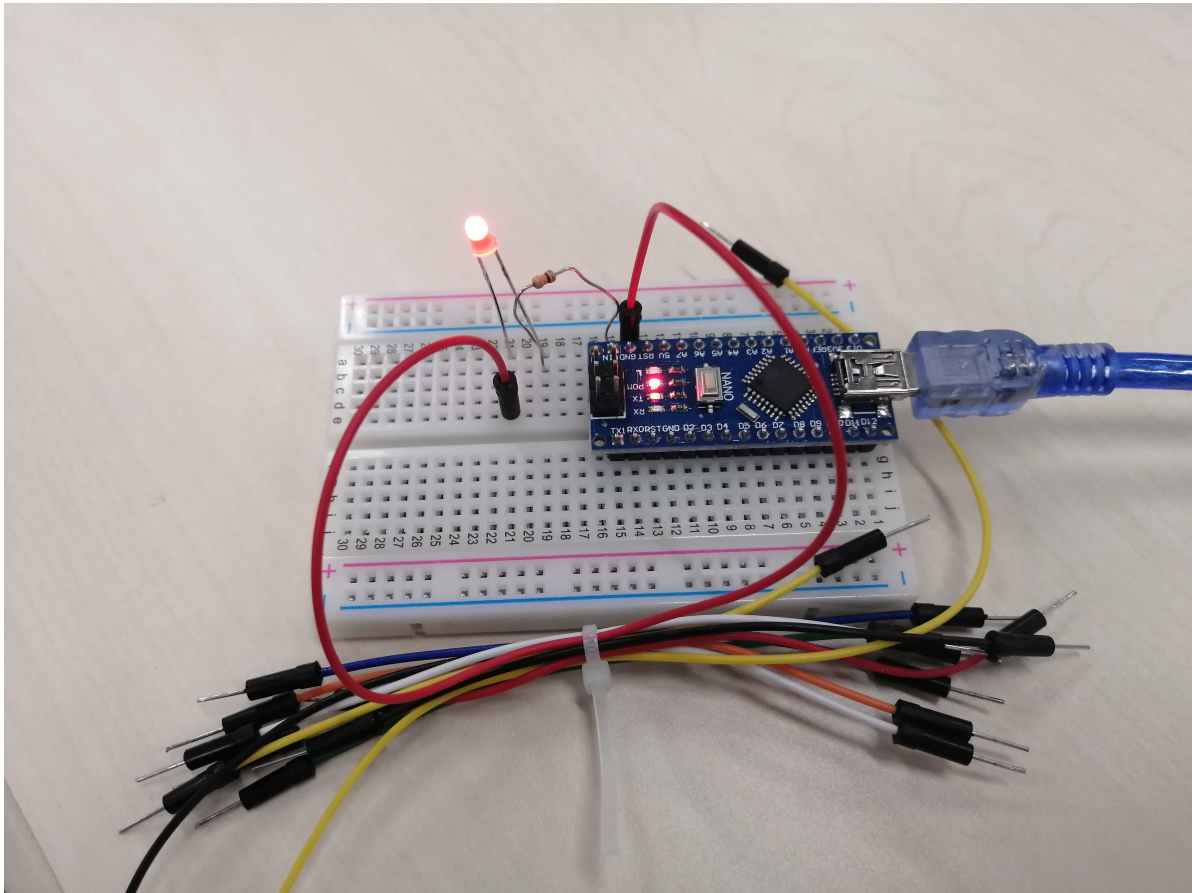


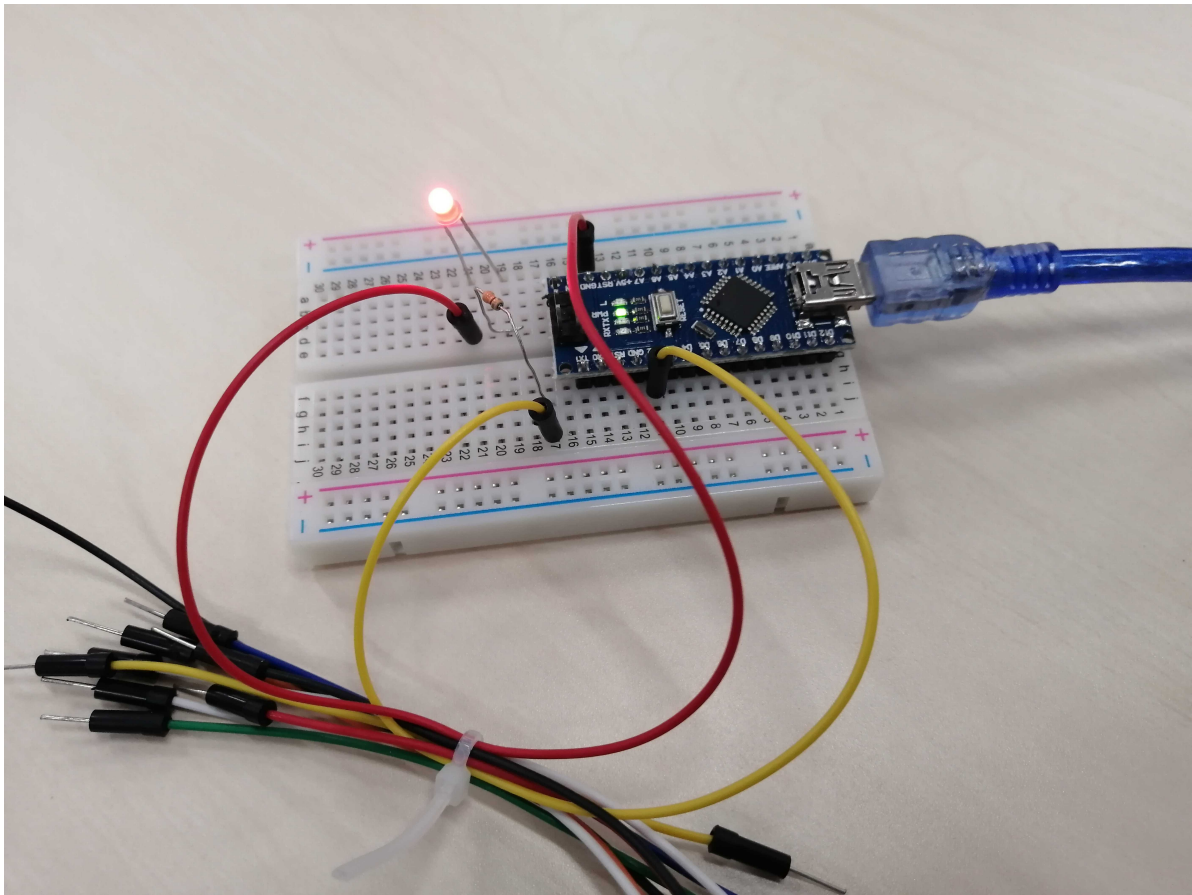
checkpoint1



checkpoint 2

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(2, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

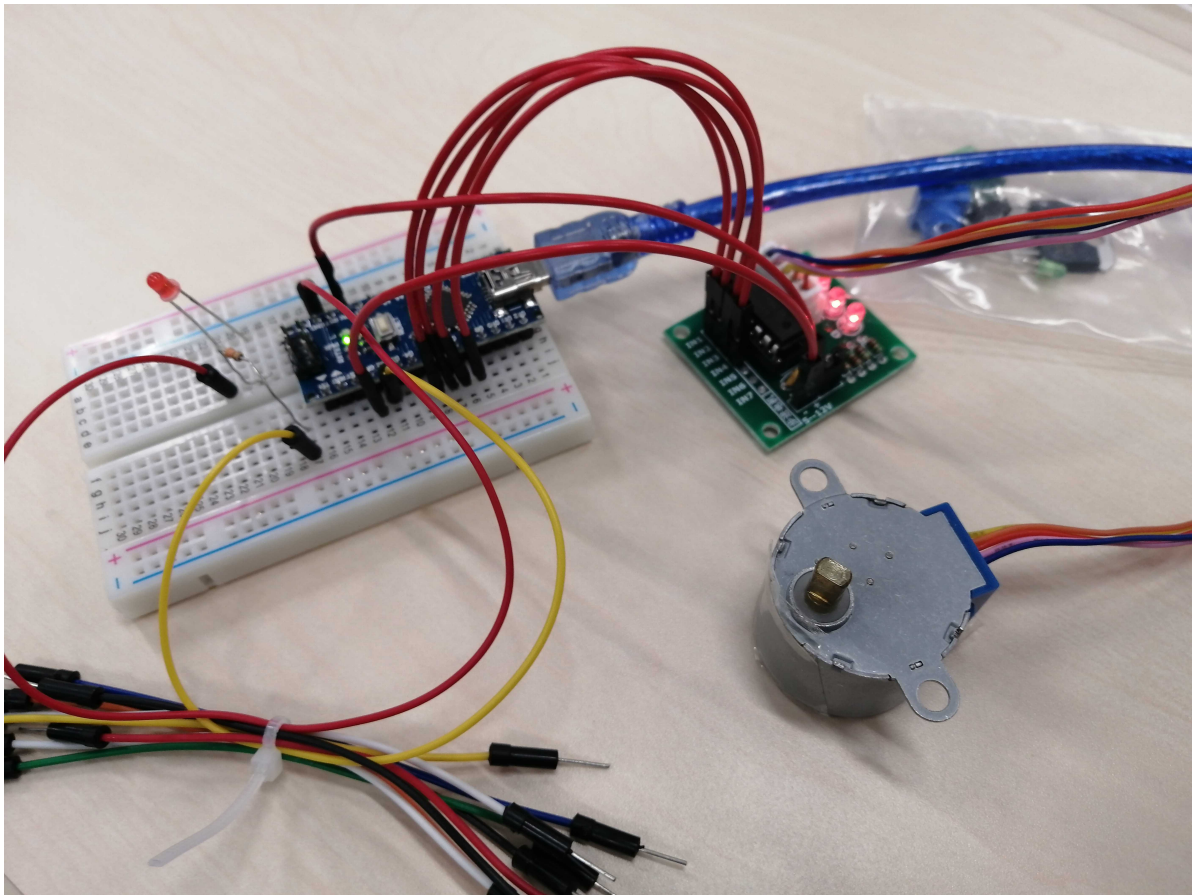


checkpoint3

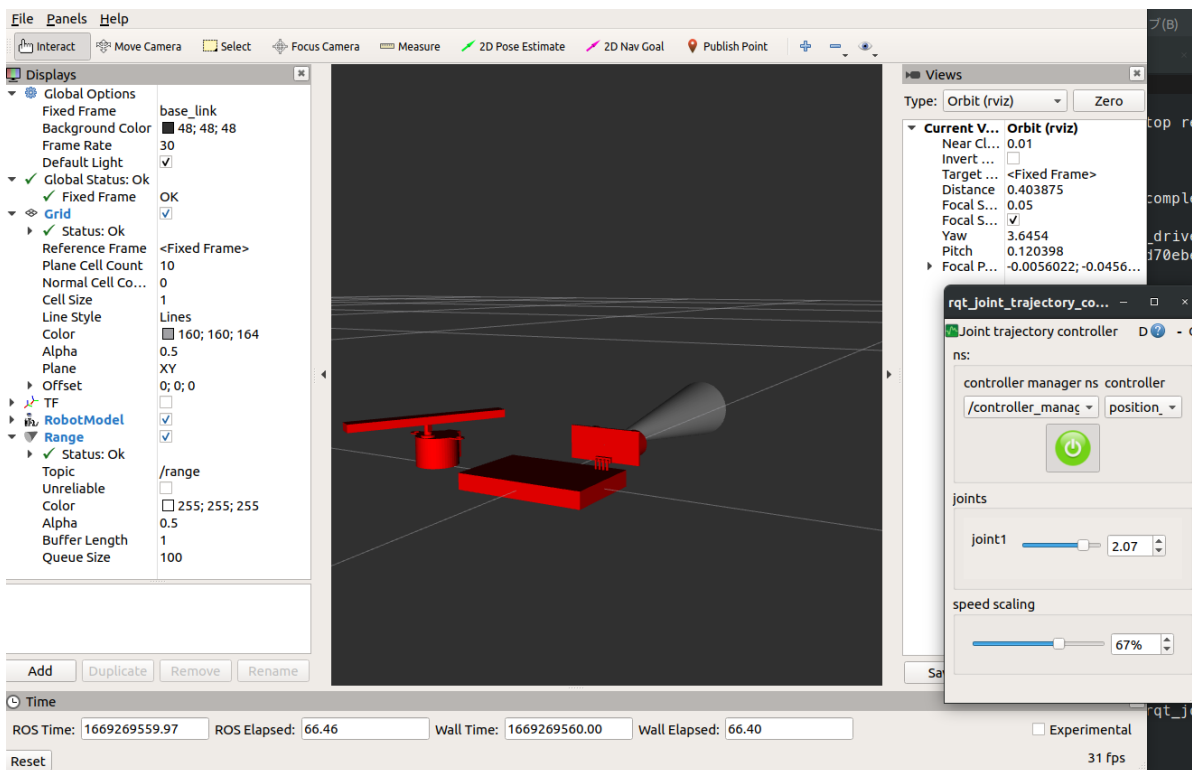
```
1  /// Stepping_motor_sample_28BYJ48.ino ///
2  #include <Stepper.h>
3  #define BAUD 9600 // シリアル通信のボーレート
4  #define MOTOR_PIN1 5 // 使用するモータの pin
5  #define MOTOR_PIN2 6
6  #define MOTOR_PIN3 7
7  #define MOTOR_PIN4 8
8  // 1 回転に必要なステップ数. 360[deg] / 5.625[deg/step] / 2(相励磁) * 64(gear 比)
9  #define STEPS_PER_ROTATE_28BYJ48 2048
10
11
12  const int StepsPerRotate = STEPS_PER_ROTATE_28BYJ48;
13  // 毎分の回転数 (rpm)
14  int rpm = 5; // 1-15rpm でないと動かない
15  // モータに与えるステップ数
16  int Steps = 512; // 90 度回転. 360deg : 90deg = 2048 : 512
17  // ライブラリとモータ配線の整合性を取り, C1, C2 を入れ替える
18  // ref https://github.com/arduino-libraries/Stepper/blob/master/src/Stepper.cpp
19  Stepper myStepper(StepsPerRotate, MOTOR_PIN1, MOTOR_PIN3, MOTOR_PIN2, MOTOR_PIN4);
20  void setup() {
21    Serial.begin(BAUD);
22    // シリアル通信の初期化
23    myStepper.setSpeed(rpm); // rpm を設定
24  }
25  void loop() {
```

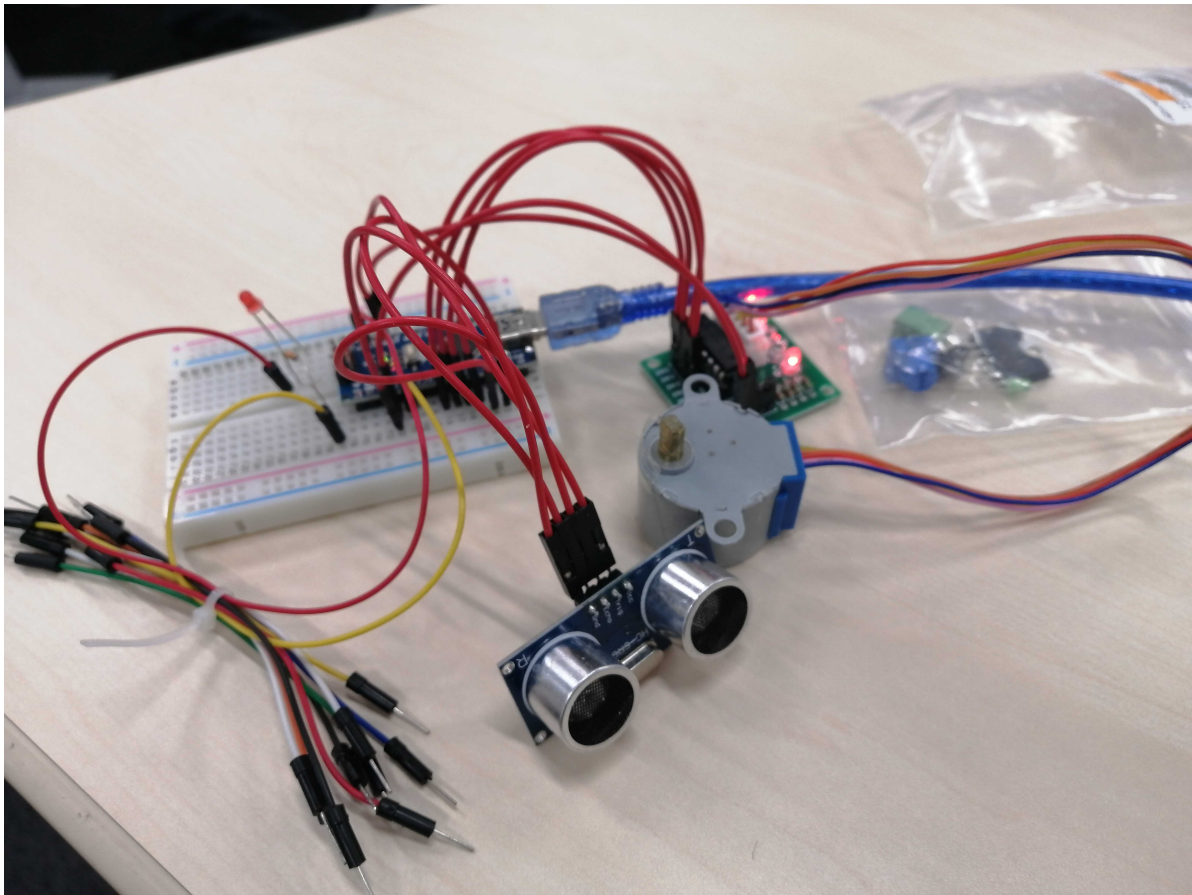
Output

最大30720バイトのフラッシュメモリのうち、スケッチが3112バイト (10%) を使っています。
最大2048バイトのRAMのうち、グローバル変数が222バイト (10%) を使っていて、ローカル変数で1826バイト使うことができます。



checkpoint4





checkpoint5

端末
端末

```

Session:/home/test3  15:06:
e-pos(x, y): (322.0, 264.0)
otor1/command 1250

e-pos(x, y): (321.0, 264.0)
otor1/command 1255

e-pos(x, y): (321.0, 263.0)
otor1/command 1260

e-pos(x, y): (323.0, 264.0)
otor1/command 1265

motor-command-by-face.l
;; face check
(if (send msg :faces)
    (progn
      (setq face (send (elt (send msg :faces) 0) :face))
      (setq *face-pos* (float-vector (send face :x) (send face :y)))
      ;; 顔がカメラの左右どちらかをチェック
      (if (<= (elt *face-pos* 0) (elt *image-center* 0))
          (setq *motor-angle* (- *motor-angle* 5))
          (setq *motor-angle* (+ *motor-angle* 5)))
      (send motor-command-msg :data *motor-angle*)
      ;; print
      (format t "face-pos(x, y): (~a, ~a)~%"
        (elt *face-pos* 0) (elt *face-pos* 1))
      (format t "/motor1/command ~a~%~%"
        (send motor-command-msg :data))
      ;; publish
      (ros::publish "/motor1/command" motor-command-msg)
    )
    (progn
      (warn "no faces~%")
    ))

```

Face detection