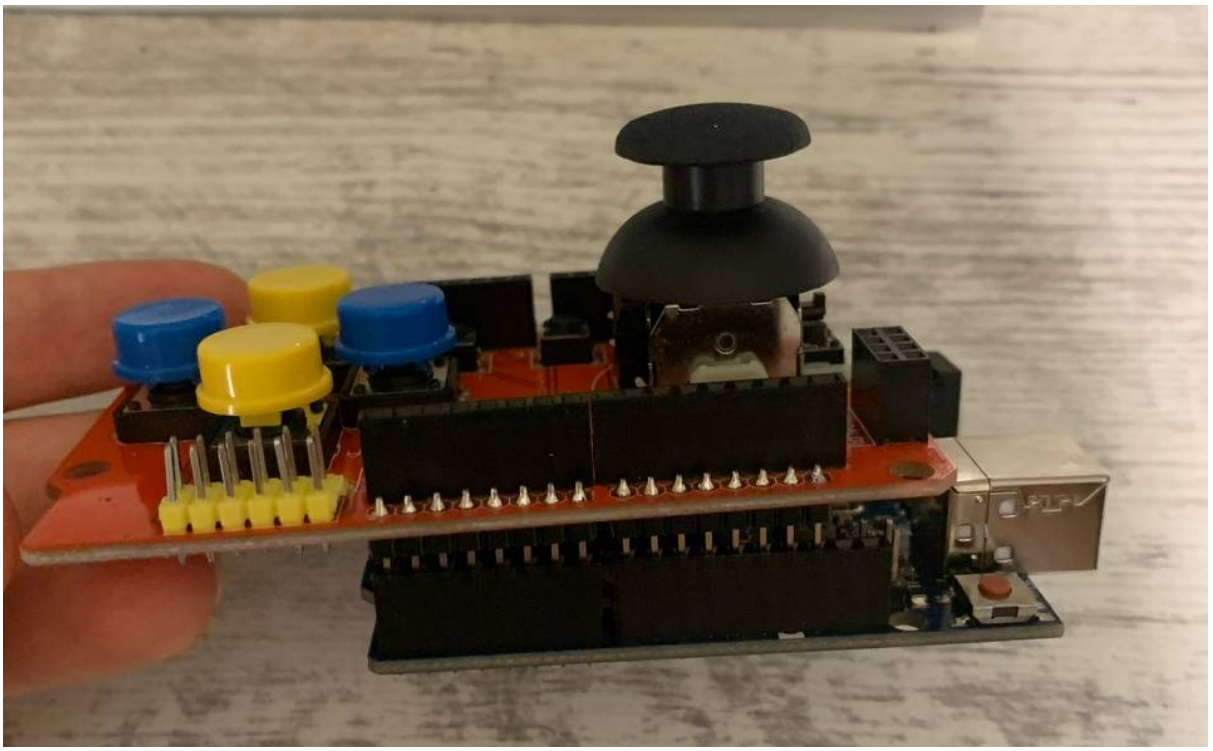
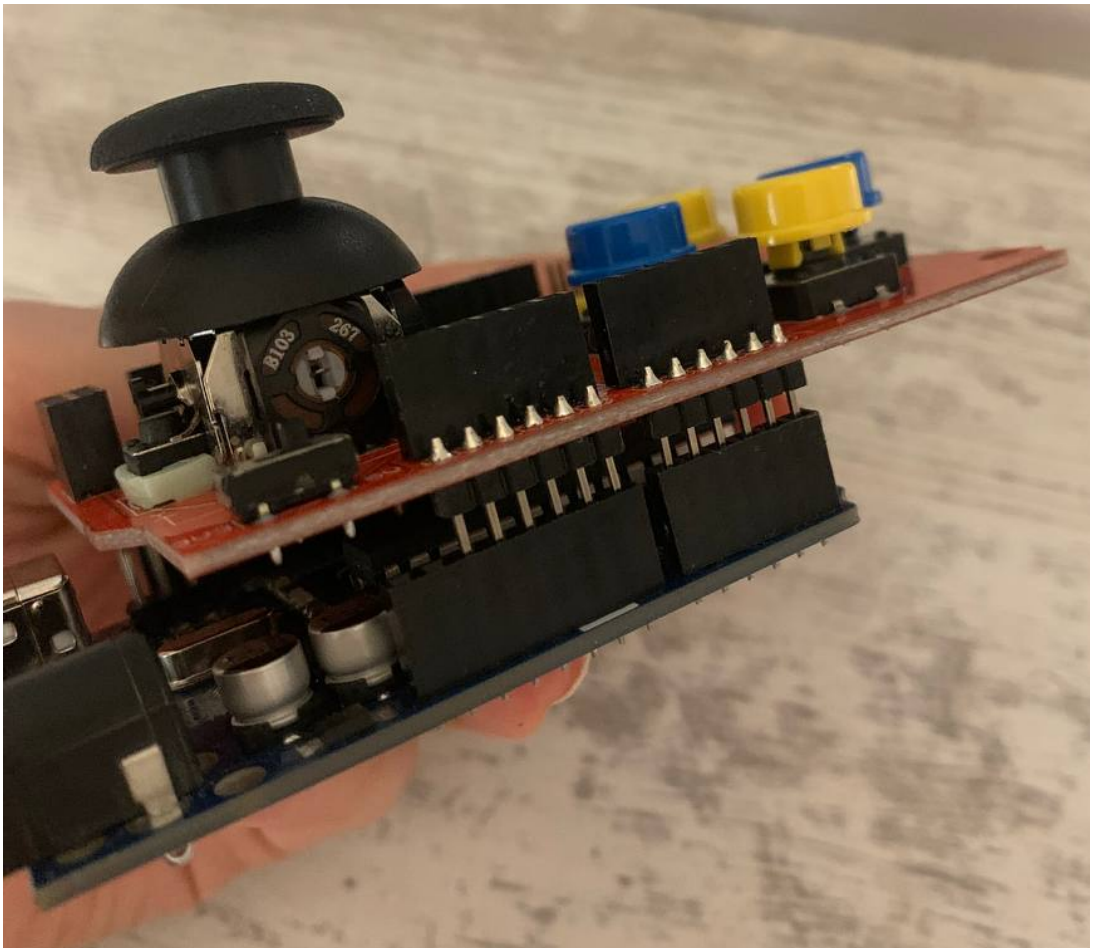
Robotic report

Made by:
Ayda Zamanian

Aug 2023

- **Joystick shield v1.A**





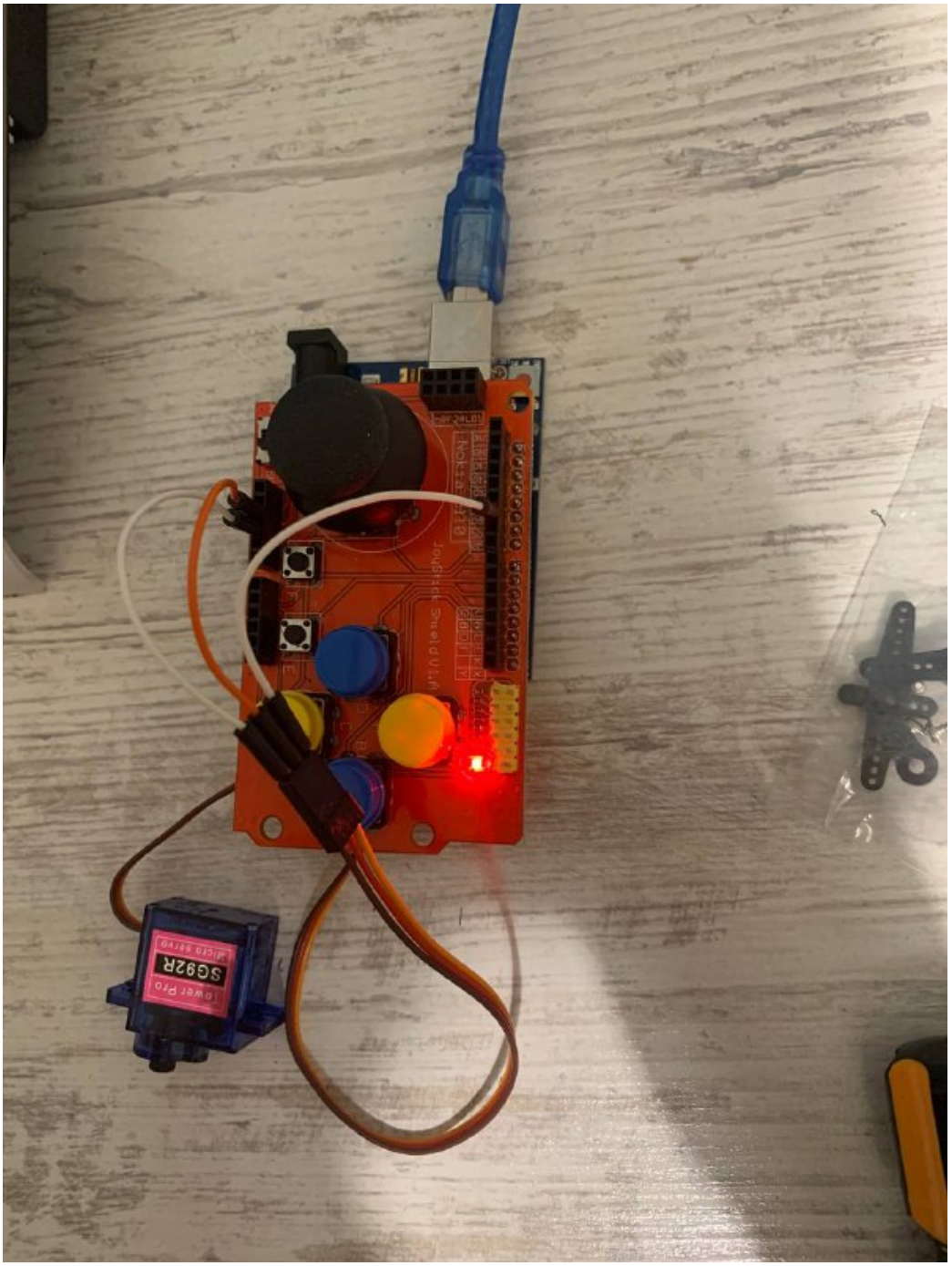
Joystick shield V1.A test (I ran it in vs code using platform io):

```
// Include the required libraries
#include <Arduino.h>
// Store the Arduino pin associated with each input
const byte BUTTON_UP = 2;
const byte BUTTON_RIGHT = 3;
const byte BUTTON_DOWN = 4;
const byte BUTTON_LEFT = 5;
const byte BUTTON_E = 6;
const byte BUTTON_F = 7;
const byte BUTTON_K = 8;
const byte PIN_ANALOG_X = 0;
const byte PIN_ANALOG_Y = 1;

void setup() {
Serial.begin(9600);

pinMode(BUTTON_UP, INPUT);
digitalWrite(BUTTON_UP, HIGH);

pinMode(BUTTON_RIGHT, INPUT);
digitalWrite(BUTTON_RIGHT, HIGH);

pinMode(BUTTON_DOWN, INPUT);
digitalWrite(BUTTON_DOWN, HIGH);

pinMode(BUTTON_LEFT, INPUT);
digitalWrite(BUTTON_LEFT, HIGH);

pinMode(BUTTON_E, INPUT);
digitalWrite(BUTTON_E, HIGH);

pinMode(BUTTON_F, INPUT);
digitalWrite(BUTTON_F, HIGH);

pinMode(BUTTON_K, INPUT);
```

```
digitalWrite(BUTTON_K, HIGH);
}

void loop() {
if (digitalRead(BUTTON_UP) == LOW) {
Serial.println("Button UP is pressed");
delay(500);
} else if (digitalRead(BUTTON_RIGHT) == LOW) {
Serial.println("Button RIGHT is pressed");
delay(500);
} else if (digitalRead(BUTTON_DOWN) == LOW) {
Serial.println("Button DOWN is pressed");
delay(500);
} else if (digitalRead(BUTTON_LEFT) == LOW) {
Serial.println("Button LEFT is pressed");
delay(500);
} else if (digitalRead(BUTTON_E) == LOW) {
Serial.println("Button E is pressed");
delay(500);
} else if (digitalRead(BUTTON_F) == LOW) {
Serial.println("Button F is pressed");
delay(500);
} else if (digitalRead(BUTTON_K) == LOW) {
Serial.println("Button K is pressed");
delay(500);
}

int xValue = analogRead(PIN_ANALOG_X);
int yValue = analogRead(PIN_ANALOG_Y);

Serial.print("X value: ");
Serial.println(xValue);
Serial.print("Y value: ");
Serial.println(yValue);

delay(500);
}
```

To connect the SG92R servo to the Arduino Joystick Shield v1.a, you will need to make the following connections:

1. Connect the VCC pin on the Joystick Shield to the 5V pin on the Arduino.
2. Connect the GND pin on the Joystick Shield to the GND pin on the Arduino.
3. Connect the VER pin on the Joystick Shield to the A0 pin on the Arduino.
4. Connect the HOR pin on the Joystick Shield to the A1 pin on the Arduino.
5. Connect the black wire of the SG92R servo to the GND pin on the Arduino.
6. Connect the red wire of the SG92R servo to the 5V pin on the Arduino.
7. Connect the yellow wire of the SG92R servo to one of the digital pins on the Arduino, such as pin 9.

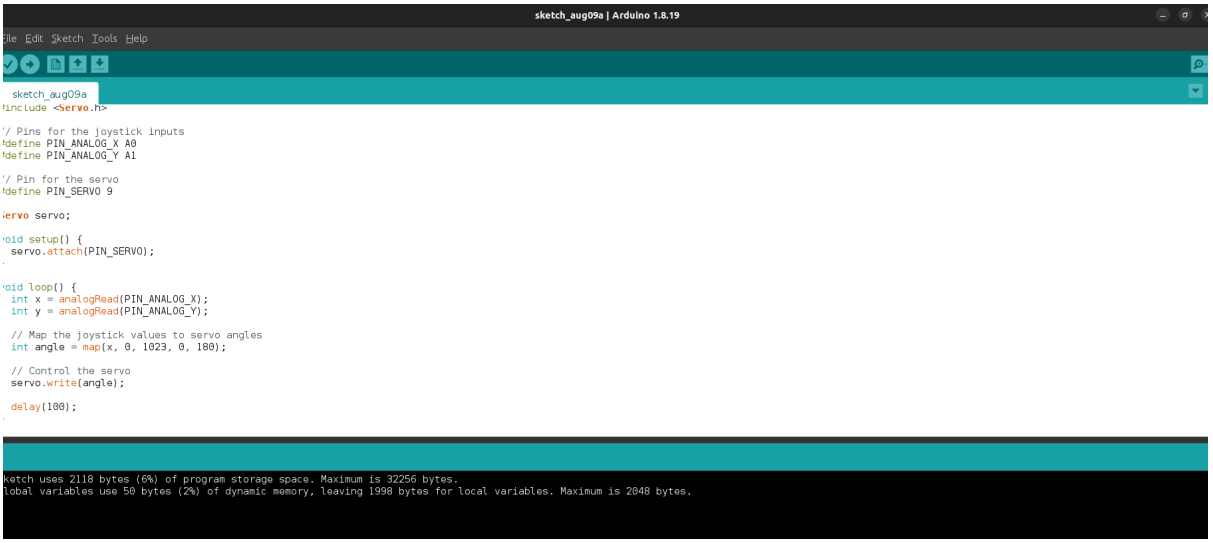


Here's an example code that demonstrates how to use the joystick inputs to control the SG92R servo:

```
#include <Servo.h>

// Pins for the joystick inputs
#define PIN_ANALOG_X A0
#define PIN_ANALOG_Y A1

// Pin for the servo
#define PIN_SERVO 9

Servo servo;

void setup() {
  servo.attach(PIN_SERVO);
}

void loop() {
  int x = analogRead(PIN_ANALOG_X);
  int y = analogRead(PIN_ANALOG_Y);

  // Map the joystick values to servo angles
  int angle = map(x, 0, 1023, 0, 180);

  // Control the servo
  servo.write(angle);

  delay(100);
}
```

This code uses the Servo library to control the SG92R servo. It reads the analog values from the joystick inputs and maps them to a servo angle using the `map()` function. The servo is then controlled using the `write()` function.

I could execute it in arduino ide



Servo library is needed : You can install it through the Arduino Library Manager.
To connect a heart pulse sensor to an Arduino and display the number on a seven-segment display, you can follow these steps:

- ## Heart pulse sensor and seven segment

1. Hardware Setup:
    - Connect the heart pulse sensor to the Arduino. The sensor will have three pins - VCC, GND, and Signal. Connect VCC to the 5V pin on the Arduino, GND to the GND pin, and Signal to an analog input pin (e.g., A0).
2. Software Setup:
    - Open the Arduino IDE and create a new sketch.
3. Code: This code assumes you have installed the necessary libraries and have defined the appropriate pin connections.

```
#include <HeartPulseSensorLibrary.h>
#include <SevenSegmentDisplayLibrary.h>

const int pulseSensorPin = A0;
const int segmentA = 2;
const int segmentB = 3;
const int segmentC = 4;
const int segmentD = 5;
// Define other segment pins as needed

void setup() {
  // Initialize the heart pulse sensor
  PulseSensor.begin(pulseSensorPin);

  // Initialize the seven-segment display
  SevenSegmentDisplay.begin();
}

void loop() {
  // Read the heart pulse sensor
  int heartRate = PulseSensor.getHeartRate();

  // Display the heart rate on the seven-segment display
  SevenSegmentDisplay.displayNumber(heartRate);
}
```