Khushi Wadhwa

Project

Write examples for official libraries.

Abstract

This project creates a comprehensive set of examples for the official Arduino libraries. Projects will range from beginner to advanced, with code and diagrams that are easy to replicate. These examples will integrate various libraries and leverage external components to facilitate an in-depth understanding of the library's functions and the Arduino as a whole.

Technical Details

In order to accomplish this project, I will code multiple examples using a variety of Arduino libraries. Examples that involve circuits and hardware will be accompanied by both fritzing diagrams and images. The examples will not be limited to using only one library, rather I will focus on implementing multiple libraries into examples to create complex projects. This will demonstrate diversity in the use of libraries and their interactions with other libraries.

Throughout Google Summer of Code, I intend to work on 6 different libraries, focusing on quality instead of quantity. I plan to spend 2 weeks on each library and submit 2-5 examples, depending on the type of library and complexity of projects. I have included details about when I plan to work on each library and potential ideas in the schedule of deliverables. The libraries I plan to work on are: * servo * capacitive sensor * keyboard * mouse * serial peripheral interface (SPI) * stepper

Previously, I have submitted a pull request to add an example to the servo library. This is example records sensory data through a potentiometer and saves it to EEPROM. The Arduino accesses the saved data and replays the actions on a servo. This example is useful for repeated actions on a servo that do not require precise measurements, simply general motions. This demonstrates how I plan to blend multiple libraries into one project, just as is done in practice. The pull request can be found here: Servo Example. I have also attached the code below.

```
 int potPin = 0;       // analog pin used to connect the potentiometer
int buttonPin = 13;   // digital pin used to connect the button
int servoPin = 9;     // digital pin used to connect the servo
int val;              // variable to read the value from the analog pin
int read;             // variable to read the data from memory

void setup() {
  pinMode(buttonPin, INPUT_PULLUP); // configure the button's pin
  myServo.attach(servoPin);   // attaches the servo pin to the Servo object
}

void loop() {
  // begin recording on potentiometer if button is pressed
  if (record == true) {
    for (int i = 1; i < EEPROM.length(); i++) {
      val = analogRead(potPin);            // reads the value of the potentiometer (valu
      val = map(val, 0, 1023, 0, 180);     // scales it to use it with the servo (value 

      EEPROM.write(i, val);                // stores data in memory
      myServo.write(val);                  // sets the servo position according to the s
      delay(15);                           // waits for the servo to get there
    }
    delay(1000);                           // pause before replay
    record = false;                        // sets record to false in order to go to rep
  }
  // begin replay on servo
  else {
    for (int i = 1; i < EEPROM.length(); i++) {
      // returns to record if button is pressed
      if (digitalRead(buttonPin) == LOW) {
        record = true;                    // sets record to true in order to go to rerec
        break;
      }
      read = EEPROM.read(i);              // sets the value of read to the value from me
      myServo.write(read);                // servo repeats the values collected from the
      delay(15);
    }
  }
}
```

Schedule of Deliverables

Community Bonding Period

- Familiarize myself with the mentors and other GSoC participants
- Discuss ideas with the team/mentors

- Adjust ideas with the feedback that I receive
- Finalize the projects and obtain approval

Phase 1 (June 1 - June 29)

- Deliverable 1
  - Week 1 - Draft 3-5 examples for the**servo** library. This includes hardware images and fritzing schematics.
  - Ideas:
    - Implement examples with ultrasonic sensors and photoresistors
    - Demonstrate uses of common objects such as buttons, switches, joysticks, and potentiometers (already included)
  - Week 2 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.
- Deliverable 2
  - Week 3 - Draft 3-5 examples (may change based on time analysis from previous week) for the **capacitive sensor** library. This includes hardware images and fritzing schematics
  - Ideas:
    - Simple dispenser system
    - Touch sensing grid
    - Mini keyboard
  - Week 4 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.

Phase 2 (July 3 - July 27)

- Deliverable 1
  - Week 5 - Draft 2-3 examples (these projects take longer) for the**keyboard** library. This includes hardware images and fritzing schematics.
  - Ideas:
    - Mini musical keyboard
    - 4x3 (or another dimension) keypad
  - Week 6 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.
- Deliverable 2
  - Week 7 - Draft 2-3 examples for the**mouse** library. This includes hardware images and fritzing schematics. Use extra time to clean up old code, double-check fritzing and hardware images, and ensure all deliverables have been met so far.
  - Ideas:
    - Integrate the Arduino and a USB mouse
    - Create a touchpad tracker
  - Week 8 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.

Phase 3 (July 31 - August 24)

- Deliverable 1
  - Week 9 - Draft 2-3 examples (these projects take longer) for the**serial peripheral interface (SPI)** library. This includes hardware images and fritzing schematics.

    - Ideas:
        - Simple 'Hello World' example
        - LED matrix/blink sequence
    - Week 10 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.
- Deliverable 2
    - Week 11 - Draft 2-3 examples for the**stepper** library. This includes hardware images and fritzing schematics. Use extra time to clean up old code, double-check fritzing and hardware images, and ensure all deliverables have been met so far.
    - Ideas:
        - Stepper controlled by joystick
        - Wheel control
    - Week 12 - Implement suggestions from mentors and ensure that code is clean. Finalize examples.

Final Week (August 24 - August 31)

- Go back and review all code for hidden errors, redundancies, and unprofessional coding practices
- Make sure all code is clean and easy to read with plentiful comments
- Check that diagrams and documentation are provided where necessary

Development Experience

Arduino was one of the first tools that I used to learn coding. My first project was a smart car with servos, an ultrasonic sensor, and a camera. It was a simple project, designed for beginners. My most recent project was a smart locking system that uses facial recognition and fingerprint technology to unlock a door. This is an example of the complexity that I plan to integrate into the Arduino libraries.

I have been coding since the 6th grade and attending hackathons since the 9th grade. Through hackathons, I have built a variety of projects and experimented with different types of technology. The most advanced project I have built is Spectra, an augmented reality application for the Microsoft HoloLens. It uses the Face API from Microsoft Azure to teach a user how to comprehend and reciprocate emotions. This project was designed for people with Autism Spectrum Disorder -- to assist individuals and ease the pressure of everyday conversations.

Because I am new to GitHub and the open-source community, I do not have a lot of projects on my GitHub profile. However, I have many projects on my Devpost: Devpost Projects. https://tinyurl.com/khushidev (Just in case the embedded link does not work).

Other Experiences

As a student at Carnegie Mellon University, I have plentiful leadership and teamwork experience. Two years ago, I founded my own hackathon -- DV Hacks. Through this experience, I learned the importance of responsibility. Not only did I have to ensure that the rest of my team completed their tasks, but I also had to keep myself on track. I realized that with, or without, someone to report to, I need to remain accountable to myself and my high standards of work. Because of this, I am both driven and hard-working in every project that I assume.

Additionally, as a Congressional Debater and Extemporaneous Speaker for 5 years, I learned a great deal about communication. Because of Speech and Debate, I find it very easy to break down complex topics and convey my ideas through words. This has been especially useful in hackathons, where I am expected to simplify my projects to the point at which a third-grader could understand them. It is this skill, and thorough understanding of my projects, that judges have always appreciated. As the captain of Extemporaneous speaking, I also realize the importance of being open to learning, regardless of position. I learn new things from novice and varsity members alike, and I am not afraid to express when I am uncertain or need feedback. Because of this, I am always open to constructive criticism to learn and grow.

Although these are not explicitly technical skills, I believe that soft skills such as teamwork, responsibility, and communication are just as important to be successful.

Why this project?

The core of the open-source community is about developers contributing to projects that are meaningful to them. As I enter into this community, I have the same goal: to give back to something that has been a catalyst in my growth. I have been working with Arduino throughout my career as a programmer and I want to contribute to this project in order to help kids like me find their passion for computer science through Arduino.

Do you have any other commitments during the GSoC period?

I may be unavailable during the week of July 13th. However, this will most likely be cancelled due to COVID-19. If it is not, I will make up the work during the down time between phases.