University of Victoria
Software Engineering
SENG 480C - Project 1

University of Victoria

Ultrasonic Movement Preceptor

Feb 7, 2022

Caleb Griffin - V00934910
calebgriffin@uvic.ca

Developer Website -
https://onlineacademiccommunity.uvic.ca/calebgriffincreativesupporttools/

About the Project

Statement of purpose for input/output design

The purpose of the ultrasonic movement preceptor system is to analyze and embody the movements of an object in a way that produces useful data to help control simple systems. In the prototype stage the movements are captured in 1D hence the use of simplistic systems. Ideally the system could be expanded into full 4D and encapsulate the entire object in order to fully replicate motion in a computer system.

The capture of the movement data in the prototype stage can be used to operate devices such as lights, lifts, and paddles. The removal of everyday I/O to operate a switch by flicking it or operating a lift by using foot pedals and buttons can all be encapsulated in this easy-to-use 1D motion capture. Levels can be set and adjusted through the distance in cm from using the device.

The full scale of the project in 4D with enhanced computer vision could provide alternatives to incredibly difficult tasks. Things such as surgeries could be reproduced and replicated from a surgeon's motions through the device. This would allow for operators to not have to be in the control room, city or even country. This would have extreme economic benefits as well increase the amount of deployable/scheduled work. Another example could be used in underwater welding where a human is at extreme risk. The system could replicate movements and allow for the weld to be done completely through robotic and computer systems while the operator uses the ultrasonic movement preceptor. This device has the specific purpose of replicating movements anywhere where the movements of an object can be modeled.

Motivation for design and intended end user's

The motivation for the project was developed through the specific question of how we can think about changing the way typical I/O operations are handled. This led to the thought of creating something using an input through motion controls. The specific tool in the Arduino kit for this type of input was the ultrasonic convertor. Any important aspect of the project is the idea of a tangible user interface and what better interface to use than the motion produced by you and everything else around you.

In our day to day lives it is getting more and more difficult to keep up with workplace demand. There are major staffing issues all over North-America, mixed with the growing concern of Covid-19 it is sometimes extremely difficult for a person to always be at their workplace. The idea for making this tool is the principle of setting the foundation to enable human and robotic interaction simultaneously. This would allow for the movement and operations of a human to be modeled through the system and duplicated anywhere in the world.

Examples on how this type of technology could grow and be utilized in the future:

1) Surgeries - With surgeon shortages around the world this technology could duplicate the movements of a surgeon's hands and model it into robotic solutions allowing a surgeon to see a patient anywhere around the world directly while using the machine. This would save time and expenses.
2) Underwater welding - A very dangerous job that many occupants lose their lives to but involves precise movement and extreme diving training. With this system a normal welder could attend to these difficult ocean welds by the movement modeled through the system in a robotic component. This could enhance safety.
3) Lift operations - when operating a lift machine in warehouse scenarios many operators need extensive training and knowledge on how to operate lifts and navigate the warehouse. This could be replaced by operators not needing to be directly in their machines and could operate them outside via the movement in the ultrasonic system.
4) Architecture planning - Scaling of models could be attached to the movement model to easily plan out on how a building would look larger or smaller
5) Any system that uses I/O can be essentially changed to the I/O of movement models

System Overview

Introduction to components and how they work

The system uses a UNO R3 Controller Board, LCD1602 Module, Ultrasonic Sensor, 830 Tie-points Breadboard, Breadboard jumper wires, USB Cable, cardboard and toilet paper roll. All of the pieces minus the cardboard can be obtained through the UNO R3 Starter Kit.
After the build (See Below) the system works by monitoring the distance in motion by sending ultrasonic waves and waiting for them to be received back. The time it takes for the waves to be received can then be converted to a readable measurement (cm). This measurement can be displayed to the LCD Monitor in real time while updating the movement of an object. For the demonstration this is modeled through the movement in a graph. The system receives power via the computer it is hooked up to so for this implementation no additional power supply is needed.

The build is limited to one ultrasonic sensor but additional sensors could be added to create additional dimensional models. The more complex the sensor model is the more data you receive. With more data to work with, more types of movement can be captured and new I/O avenues can be obtained.

Arduino Build

Components Needed
(All components can be found in “The Most Complete Starter Kit UNO R3 Project”)

UNO R3 CONTROLLER BOARD - 1PC

ULTRASONIC SENSOR - 1PC

LCD1602 MODULE - 1PC

BREADBOARD JUMPER WIRE - 18PC
BLACK - 5PC
RED - 4PC
YELLOW - 5PC
ORANGE - 4PC

830 TIE-POINTS BREADBOARD - 1PC

USB CABLE - 1PC

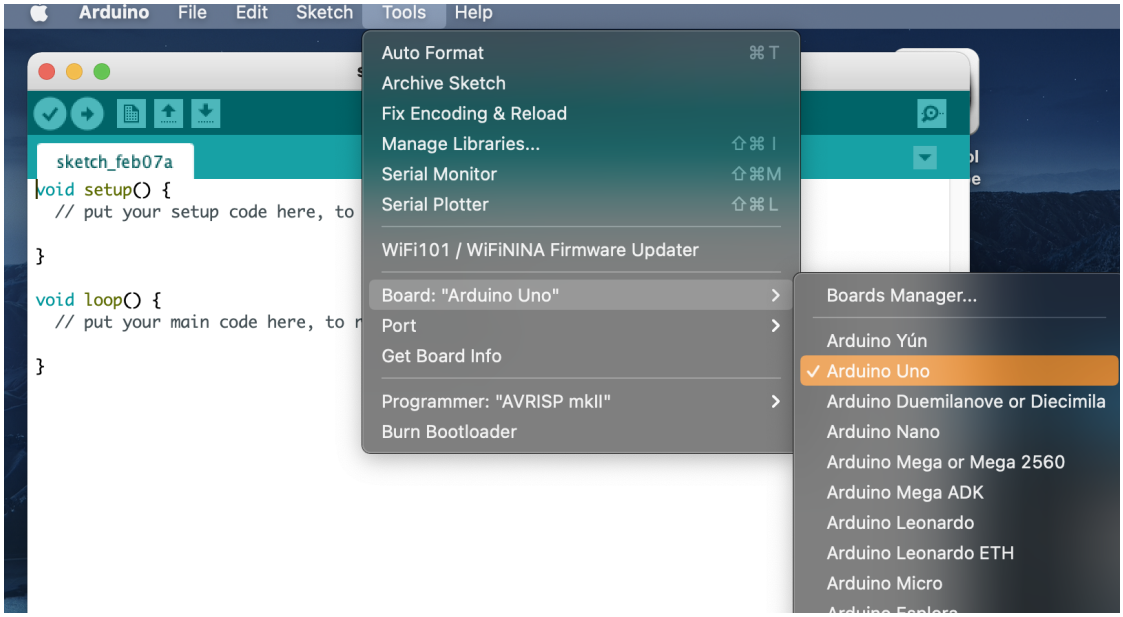
CARDBOARD - 2PC

EMPTY TOILET PAPER - 1PC

Before Getting Started

Initially before getting started you should have the Arduino software downloaded. You can find the latest version at this link: https://www.arduino.cc/en/software

Once the software is downloaded, load up the program. You can now plug in the Arduino board via the USB cable. Getting your environment set up properly is essential for uploading the program onto the board. In the Arduino program select Tools ->Board -> Arduino Uno (Image below)



You will also need to make sure that the proper port is selected and similarly this can be done in the same way by going Tools->Port->(Select Port) (Image below)

You are now ready for the wiring and build of the system.

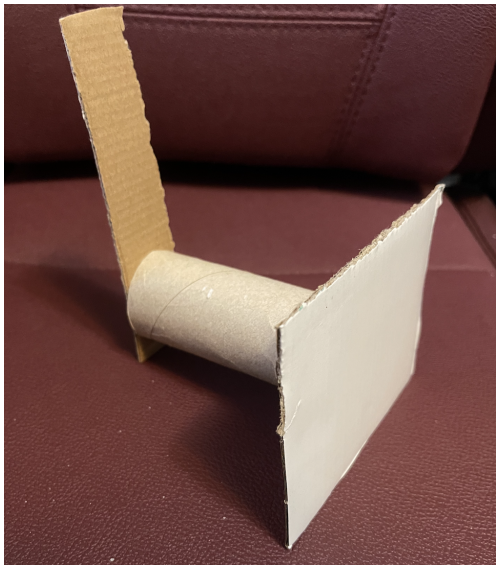
Wiring and Setup

*note that the color of the cords does not matter but are used for organizational purposes only
Please follow along to the directions below for setting up the systems wiring and components.

CORD	FROM	TO
Black -	J1 (breadboard)	Rail(-) (breadboard)
Red +	J2 (breadboard)	Rail(+) (breadboard)
Black -	J3 (breadboard)	Rail(-) (breadboard)
Yellow	J4 (breadboard)	11 (UNO Board)
Yellow	J5 (breadboard)	10 (UNO Board)
Yellow	J6 (breadboard)	9 (UNO Board)
Orange	J11 (breadboard)	2 (UNO Board)
Orange	J12 (breadboard)	3 (UNO Board)
Orange	J13 (breadboard)	4 (UNO Board)
Orange	J14 (breadboard)	5 (UNO Board)
Red +	J15 (breadboard)	Rail(+) (breadboard)
Black -	J16 (breadboard)	Rail(-) (breadboard)
Red +	D60 (breadboard)	Rail(+) (breadboard)
Yellow	D61 (breadboard)	13 (UNO Board)
Yellow	D62 (breadboard)	12 (UNO Board)
Black -	D63 (breadboard)	Rail(-) (breadboard)
Red +	3.3V (UNO Board)	Rail(+) (breadboard)
Black -	GND (UNO Board)	Rail(-) (breadboard)

Ultrasonic Sensor (breadboard) - B60,B61,B62,B63
LED Module (breadboard) - F1,F2,F3,F4,F5,F6,F7,F8,F9…F16

Additionally the push object can be made from a empty roll of toilet paper and gluing two more pieces cardboard to either end as shown below:



You should now be ready to program.

Code + Program Execution

The code to this project is an adaptation of the open source publication of “Simple Ultrasonic Distance Measure With LCD Display" from user onatto22 on hackster.io. Additionally components of Ahmed Djebali implementation on the use of ultrasonic sensors were taken into account. Credits are given in the code.

For the code implementation, you must include the LED library for the simplest working solution. You can do this by (#include <LiquidCrystal.h>) above the setup loop. Additionally you must declare where the trigger and echo pins are for the ultrasonic sensor.

Inside the setup loop you must turn on the pins for the ultrasonic sensor, as well you must turn on and run the LCD display. Lastly you must initialize the program to allow it to write to the graph plotter.

For the void loop you must declare a duration and distance variable, then measure the length of the trigger to echo and convert it to whatever measurement type you are using. For this implementation I use cm. Lastly you must print the conversion to the LED and write to the graph plotter to visualize the movement.

Refer to the code below for my implementation:

A1_480C §

```
/*
Caleb Griffin
V00934910

Credit for help on implementation
Ahmed Djebali - on use of how ultrasonic sensors work
onatto22 - on initializing LCD
*/
#include <LiquidCrystal.h>            //Adding Liquid Crystal Library
LiquidCrystal LCD(11,10,9,2,3,4,5);

#define trig 13 //Sensor Setup
#define echo 12

void setup()
{
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);

  LCD.begin(16,2);               //Turns on LCD
  LCD.setCursor(0,0);            //Set LCD cursor to upper left corner
  LCD.print("Distance Level:");
  Serial.begin(9600);
}

void loop() {
  long duration, distance;
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = (duration/2) / 29.1;     //cm conversion

  if (distance>20){ .                // Setting max bounds to 20cm (can be changed)
    distance = 20;
  }

  Serial.println(distance);

  LCD.setCursor(0,1);
  LCD.print("                ");
  LCD.setCursor(0,1);
  LCD.print(distance);
  LCD.print("cm to front");
  delay(250);

}
```