

Project Documentation

Project Title: e-Glove

Team Members: Harshit Rathore, Shubham Agrawal & Elle Atma Vidya Prakash

Team Mentors: Rohit Agarwal & Divya Prakash

Basic aim:

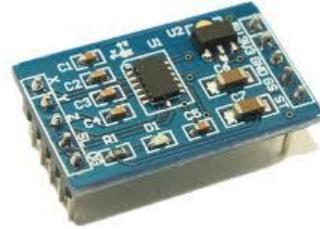
To make a glove embedded with various sensors to detect hand and finger gestures, and implement those in many things, like playing games, giving presentations and many more.

Motivation: We were having a look at previous year projects done under the electronics club, we saw many game controllers. So we thought of a game controller that can work for all the latest games, is full featured, and gaming looks more realistic using that. Inspired by things like Microsoft© Kinect™, Sony© Play Station™ motion controller, and many more such controllers available in the market, we came up with this idea.

Theory:

Sensors

1. Accelerometer: An accelerometer is a device that measures acceleration. When it is kept horizontal at rest, it measures 9.8 N/Kg downward. Whenever there is a tilt, a small component is left at the downward face, which can be detected.



2. Flex Sensor: The flex sensor is basically a variable resistor that reacts to bends. It changes its resistance when flexed so we can measure that change. The bend is only detected in one direction. TO read the data from the sensor, we need a fixed resistor (not changing) that we can use for that comparison (We are using a 33K resistor). This is called a voltage divider and divides the 5v between the flex sensor and the resistor.



Microcontroller

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. The one we used is **Arduino Uno**.

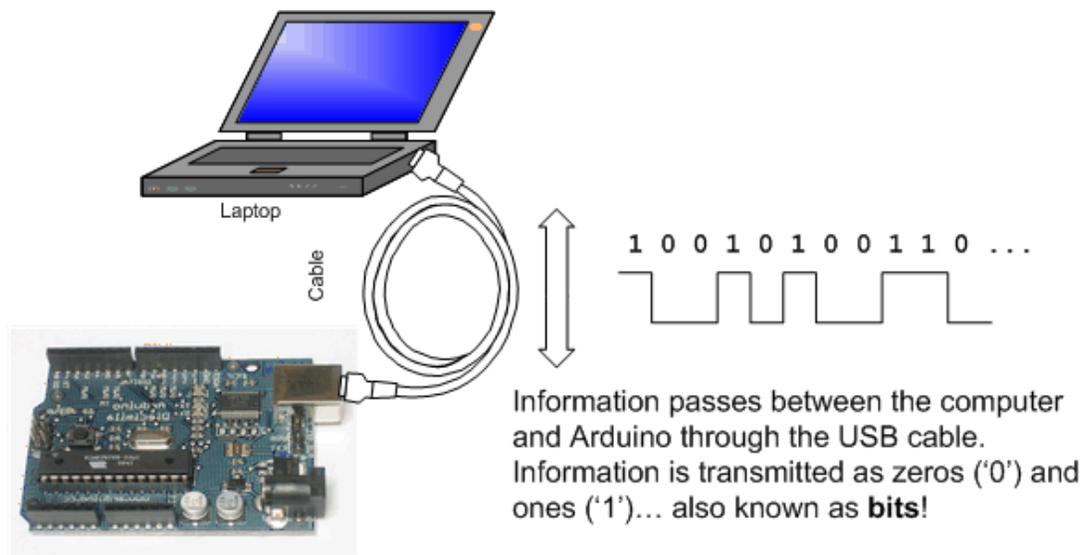
Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller. An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields.



The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Communication

We have used Serial communication (also known as UART) to send data from arduino to the computer through USB serial COM ports.



UART: A Universal Asynchronous Receiver/Transmitter, abbreviated UART, is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. We used UART to transfer our data from arduino to the computer through serial COM ports for further processing.

DirectX:

Microsoft DirectX is a collection of application programming interfaces (APIs) for handling tasks related to multimedia, especially game programming and video, on Microsoft platforms.

In late 1994, Microsoft was ready to release Windows 95, its next operating system. Before this, DOS was used. DOS allowed direct access to video cards, keyboards, mice, sound devices, and all other parts of the system, while Windows 95 - with its protected memory model - restricted access to all of these, working on a much more standardized model. Microsoft needed a way to quickly let programmers get a solution, and they came up with DirectX. DirectX allowed all versions of Microsoft Windows, starting with Windows 95, to incorporate high-performance multimedia. The functions used for programming the DirectX are included in windows library.

Programming Platforms:

For programming the arduino, we have used the default arduino IDE.

The code running on the computer which receives the data from serial port, and decodes it to perform various functions is written in C.

The C compiler used is DEV C++.

Implementation:

- ✚ All the components are giving readings to the arduino, where we are converting these readings into character code in order to send the bytes efficiently and quickly to

the computer. Inside a never ending loop, every reading corresponds to a character. So a package is created every time the loop runs, and that is sent to the computer. This package contains the data coming from each and every sensor, and finally this needs to be decoded on the computer to get the readings.

- ✚ A code should run on the computer in order to receive the data that is being sent by arduino via serial ports. This code is written in C language. The function *com->connect()* initializes the serial communication between the computer and arduino. We have written a library in which we have defined this function. The connection is refreshed on starting of the loop so that the values do not get mixed up.

- ✚ The character code is decoded according to the algorithm, and now the data is used to control the mouse and the keyboard. The functions to move the mouse and pressing the keys of the keyboard are defined in windows library, windows.h.

- ✚ But this movement won't work inside games as games take data directly from the hardware, not from the windows. The application utility required for this purpose is known as **DirectX**, which most of the modern games require. DirectX supplies the hardware input data

directly to the games for high speed data rate. The **SendInput()** function feeds the data to DirectX.

Final Working:

The gestures implemented are-

- ✚ Tilting of palm in 4 directions (about 2 axes).
- ✚ Tilting of wrist in 2 directions (about 1 axis).
- ✚ Bending of thumb, index finger, middle finger and pinkie finger.
- ✚ Bending of wrist horizontally.

Summing it all up, we are able to control 12 different things at a time, and all of them can work simultaneously without any delay. We have controlled 4 directions of the mouse, 4 direction keys of the keyboard, mouse left and right click, control key and space bar. All these can be reconfigured to any key of the keyboard by slight modifications in the code.

Utility

- ✚ Gaming



✚ Easy Presentation



Further development and future scope

- ✚ Making the glove Wireless using X-Bee or Bluetooth.
- ✚ Using an IMU to get directions easily and more accurately for all the motions of the hand.
- ✚ Implementing more gestures using the current hardware.
- ✚ Implementing plug and play, by making a driver for the hardware.

Useful links

DirectX programming information <http://www.goo.gl/gsXle>

Programming related queries <http://stackoverflow.com/>

C programming Tutorials <http://www.cprogramming.com/tutorial/c-tutorial.html>

A word of thanks

We would like to thank our mentors Divya Prakash and Rohit Agarwal, and Coordinators Shivendu Bhushan, Sonu Agarwal and Swapnil Upadhyay for their guidance, patience, suggestions and their belief in us .They inspired us to learn a lot and work on this exciting project and checked the progress of our project so sincerely that even if , sometimes, when we were like, frustrated and hopeless regarding our project they showed us the way.

Moreover, it was fun doing this project in first years' summers as we got to learn so much.