9 Peg Hole Game

for rehabilitation of stroke affected patients

Motivation

Rehabilitation for the recovery of an illness like stroke is an expensive and time taking process involving physio-exercises under the supervision of a doctor. In our country, this generally goes neglected as many people are unable to avail these facilities due to monetary reasons and the recovery becomes more time taking than it should be because of sheer neglect.

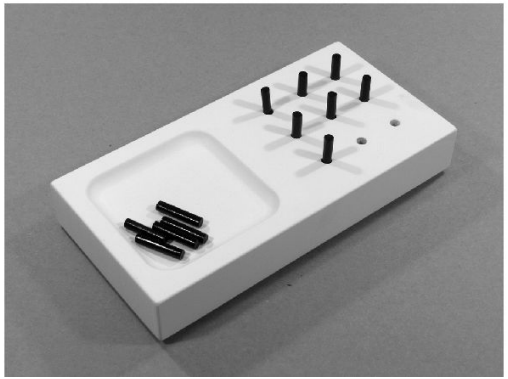
The aim of our project is to design and construct an equipment which would essentially help a patient in performing one such exercise which helps in the recovery of upper limb.

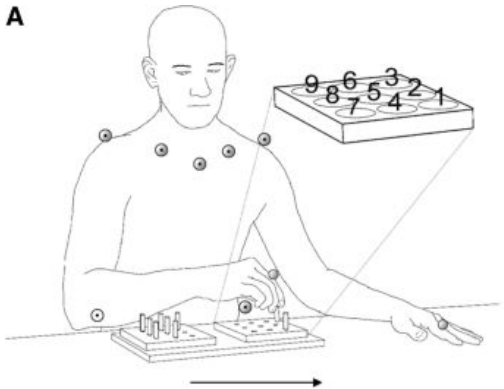
The Nine-Hole Peg Test is used to measure finger dexterity. It being a somewhat “competitive game” with corresponding scoring would make the patient more enthusiastic about the exercise and stimulate a slightly more paced recovery.

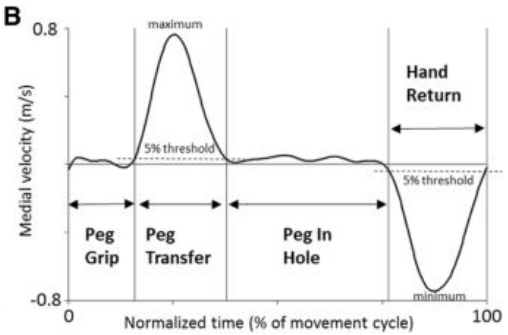
How it works

The model consists of 9 3x3 grid of holes, equally spaced from each other which are to be filled by 9 pegs that are provided. The motive of the game is for the patient to place the pegs accurately into the holes in less time as possible.

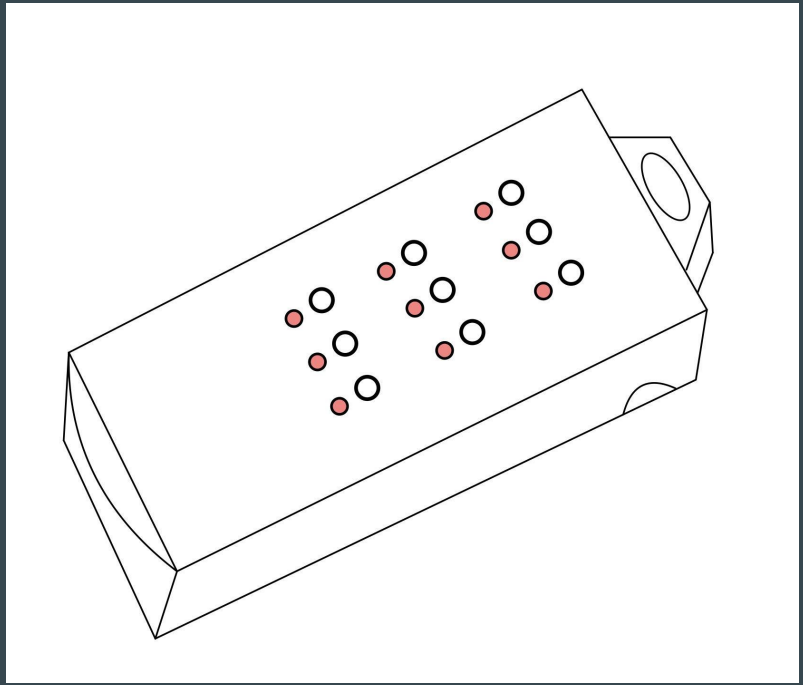
The hardware implementation of the game is inside the 3D box, hidden from the user, making the experience user friendly. LED lights, buzzer, and a display for timer are specifications attached to make the game interactive.







Physical Model



- 3D Printed Model
- Pegs
- Leds
- Dashboard
- TimeDisplay
- Model Status
- Sensor Readings

Our Functionalities

- **3D Printed Model**

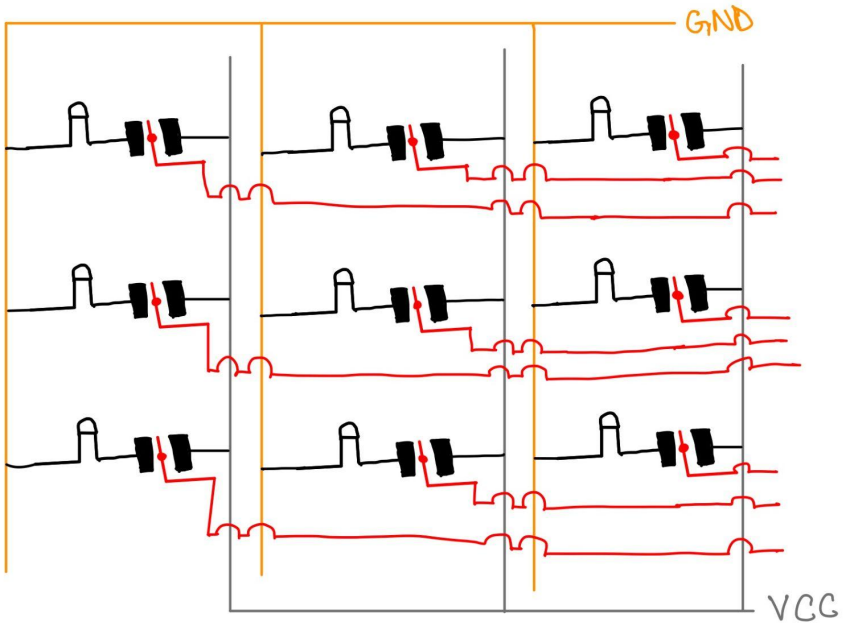
- **Pegs**
  The pegs are to be inserted in the holes of the 9 Peg Hole Test

- **LEDs**
  The corresponding LEDs glow whenever a peg has been inserted into a hole.

- **Sensor Readings**
  We upload graphs of the EMG Sensor and GSR Sensor on our website for the doctor to view

- **Dashboard**
  It portrays informations such as the sensor data and the led state. It can also track the time taken by the patient

Circuit Diagram



Implementation

The pegs are to be placed in the holes of the 9 Peg Hole Game. Whenever a peg is placed in a hole, the corresponding circuit is completed and the respective LED glows.

The circuit is initially broken. Whenever a peg is placed in the hole, with the help of conducting sheet which is attached to ground and vcc, the circuit is completed and the corresponding LED glows.

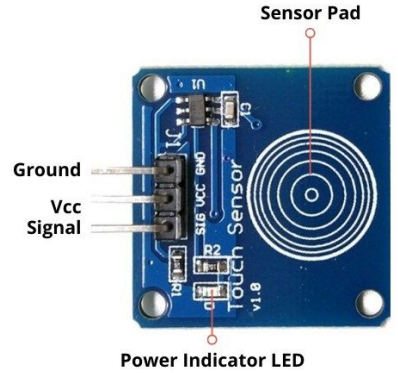
The current status of the holes are displayed on the website i.e if a peg has been placed in a hole, the corresponding hole in the website glows. This helps the doctor virtually see the progress made by a patient.

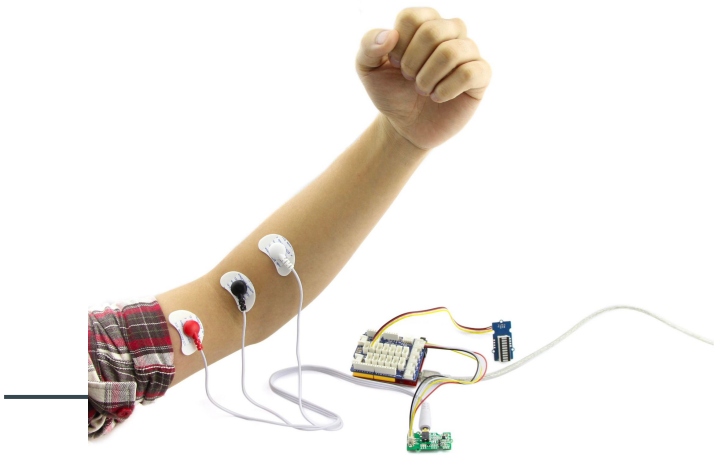
The timer is started when the patient touches the first peg and once all pegs are in the holes the timer is stopped.

Sensors

The following sensors are used for implementing the game:

1. EMG Sensor
   The electrodes of the EMG Sensor detect electrical activity generated by muscle movement/contraction.
2. GSR Sensor
   It is used to measure the electrical conductance of the skin. This sensor allows us to detect strong emotions which cause stimulus to the sympathetic nervous system.





Data Analytics

We take the data of when which peg is being inserted and can talk about the stronger sides of the patient along with EMG and GSR values

Thingspeak

We get the value of EMG Sensor and GSR sensor which is uploaded on thingspeak.

We also upload the values of the LEDs to know the current status

The following libraries were used:

- WiFi.h
- ThingSpeak.h

Challenges Faced

The hardware part of the project was the biggest challenge of all. Making a responsive and sensitive version of the 9-peg hole game and maintaining the compact structure and functionality of the game was truly difficult.
However, we overcame most of the difficulties by using the conducting sheet. We modified the peg in a very innovative way to cater to our needs.

Even the onem2m connection was a hassle

Our Contributions

Anushka:
Hardware, Sensors, and Thingspeak
Himani:
Hardware, Presentation, Dashboard, oneM2M
Nandini:
Hardware, Sensors
Anuhya:
Hardware, Website, and oneM2M