NTX Presentation 2024

Neurotech UofT

The problem

- About 15% of the world's population lives with some form of disability, of whom 2-4% experience significant difficulties in functioning
- Among this statistic are individuals with weakened grip strength or hand amputees
- Weakened grip strength or inability to perform grasping motions strongly limit the ability to perform daily household tasks

Our Design

Our aim is to create a prototype of a mind controlled prosthetic hand that can conduct hand grasping motions based on the user's thoughts

Objectives

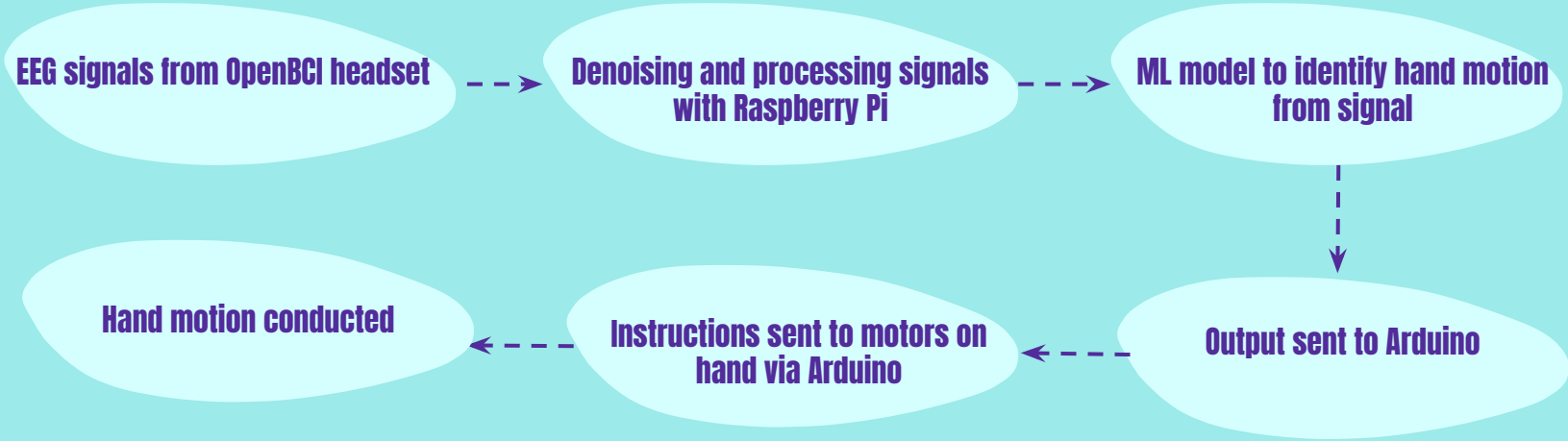
1) Create a prosthetic hand that can conduct the following two motions:

- Opening hand
- Closing hand

2) Identify the above two hand motions from EEG signals

3) Retrieve relevant EEG signals from OpenBCI headset

Pipeline

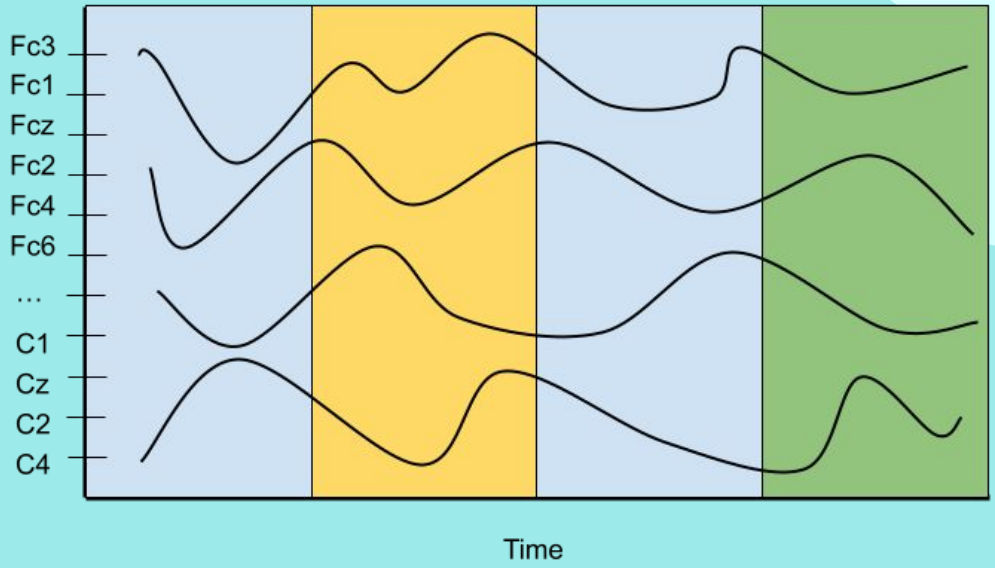

EEG signals from OpenBCI headset
Denoising and processing signals with Raspberry Pi
ML model to identify hand motion from signal
Output sent to Arduino
Instructions sent to motors on hand via Arduino
Hand motion conducted

Software subsystem

Data Processing and Machine Learning

Identifying and understanding training data

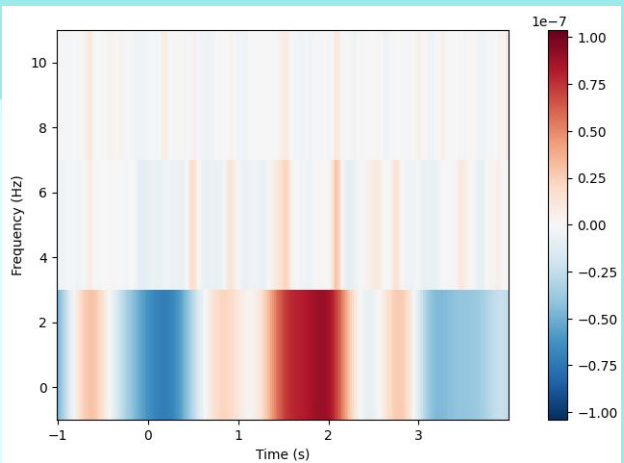
- Training data from “EEG Motor Movement/Imagery Dataset” by Schalk, G. et. al.
- 109 subjects subject to visual stimulation (target on a screen) and told to either close or imagine closing left fist, right fist, both fists or feet at sight of stimuli
- 2 baseline rest trials, 12 trials of 4 second periods of rest, action, rest, etc. for 2 minutes

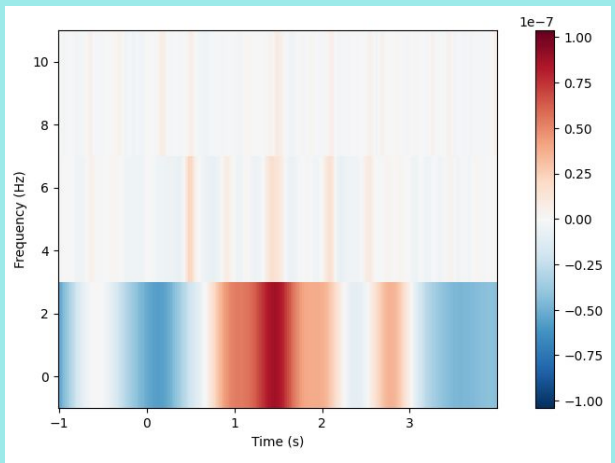


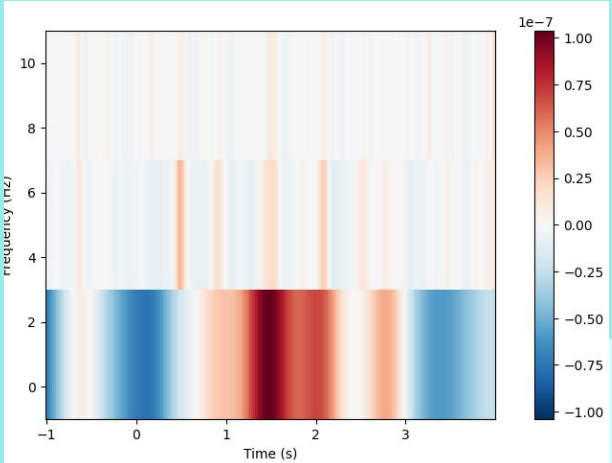
Channel and Frequency investigation

- Used channels C3, C4, and Cz (as denoted by literature to work best for motor imagery tasks)

- Conducted EEG power analysis to determine most active frequencies. Frequencies used:
  - 1-4 Hz: Increased activation 2 seconds after hand closure
  - 13-17 Hz: Sharp peak observed on the onset of hand closure

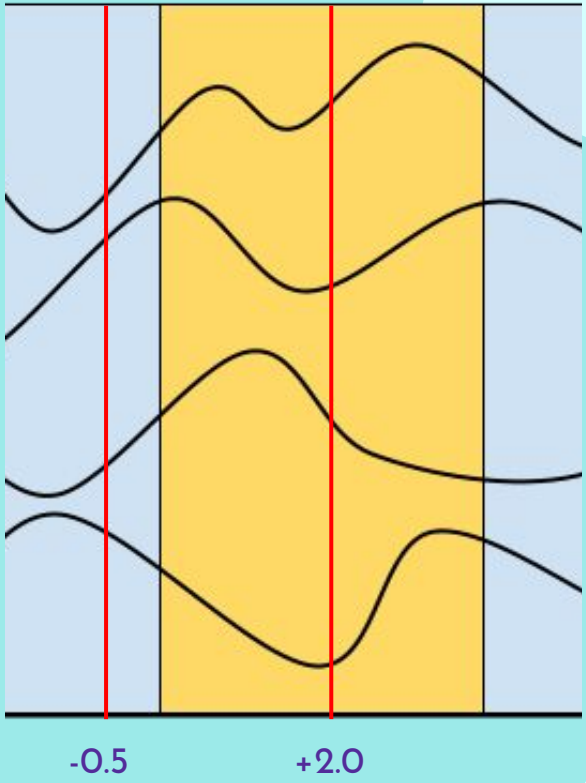






Data preprocessing

- Treat imagined and executed movements of same variety as same event
  - Ex. imagined left hand closing and executed left hand closing treated as same event
- Created two sets of data:
  - Bandpass filter of 1-4Hz
  - Bandpass filter of 13-17 Hz, followed by baseline correction.
- Epoched all points of interest from -0.5 to 2.0 seconds relative to initial event
- Averaged epochs at each trial of the same label
  - Ex. For subject 1, all left hand closing events for trial 3 would be averaged



Ground truth label assignment

- Disregarded all events related to closing feet
- All events that were related any hand closing (left, right, or both) given label of 1
- All rest events given label of 0

Picking an ML Model: Why we Chose Temporal Convolutional Networks (TCNNs)

TCNNs over RNN/LSTM Architecture

- We faced issues with vanishing and exploding gradients when trying to train an RNN/LSTM based model

- It is widely accepted that RNN/LSTM architectures take longer to train than CNNs

- TCNNs have seen success in EEG classification tasks such as motion imagery (Lun et al., 2020)

Our Model Architecture

Important Notes:

- Significant dropout (80%) to reduce overfitting to the dataset
- He initialization in convolutional layers to deal with vanishing/exploding gradients (He et al., 2015)
- A large number of filters (32, 64) in the 1D convolutional layers to learn a richer set of features from the EEG input data

```
def create_model():
    dropout = 0.80

    input_layer = keras.Input(shape=(401, 3))

    x = Conv1D(filters=32, kernel_size=3, strides=2, activation = 'relu', kernel_initializer=HeNormal(), padding="same")(input_layer)
    x = BatchNormalization()(x)
    x = MaxPooling1D(pool_size=2, strides=2)(x)


    x = Conv1D(filters=64, kernel_size=3, strides=2, activation = 'relu', kernel_initializer=HeNormal(), padding="same")(x)
    x = BatchNormalization()(x)
    x = MaxPooling1D(pool_size=2, strides=2)(x)

    x = Flatten()(x)

    x = Dense(1024, activation="relu", kernel_initializer=HeNormal())(x)
    x = Dropout(dropout)(x)

    x = Dense(256, activation="relu", kernel_initializer=HeNormal())(x)
    x = Dropout(dropout)(x)
    output_layer = Dense(1, kernel_initializer=HeNormal(),activation="sigmoid")(x)

    return keras.Model(inputs=input_layer, outputs=output_layer)

model = create_model()
model.summary()
```

Model Training

```
from tensorflow.keras.models import load_model
model = load_model('model.keras')
checkpoint_filepath = '/content/drive/MyDrive/NeuroTechUofT/model.keras'

callbacks = [
    keras.callbacks.ModelCheckpoint(
        checkpoint_filepath, save_best_only=True, monitor="binary_accuracy", mode="max",),

    keras.callbacks.ReduceLROnPlateau(monitor="val_binary_accuracy",factor=0.9,
                                      patience=3, min_lr=0.00000000000001,),
]
optimizer = keras.optimizers.Adam( learning_rate=1e-5)
loss = keras.losses.BinaryCrossentropy()
epochs = 50
model.compile(
    optimizer=optimizer,
    loss=loss,
    metrics=[
        keras.metrics.BinaryAccuracy(),
        keras.metrics.AUC(),
        keras.metrics.Precision(),
        keras.metrics.Recall(),
    ],
)
model_history = model.fit(
    train_dataset,
    epochs=epochs,
    callbacks=callbacks,
    validation_data=test_dataset,
    class_weight=class_weight
)
```

What we tried:

- Hyperparameter tuning with the learning rate, learning rate schedule, and number of epochs
- A learning rate schedule for stable convergence to local optima
- Repeatedly training the model for a short number of epochs while changing hyperparameters to improve performance

Metrics

Work	Number of electrodes	Number of MI tasks	Database	Global averaged accuracy (%)	Methods
Kim et al. (2016)	14	4	Physionet	77.70	SUTCCSP
Pinheiro et al. (2018)	2	4	Physionet	74.96	RNA
Dose et al. (2018)	64	4	Physionet	80.38	CNN
Karácsony et al. (2019)	64	4	Physionet	76.37	CNN
Hou et al. (2019)	–	4	Physionet	95.54	ESI+CNN
This work	2	4	Physionet	97.28	CNN

Validation accuracy for various models motor imagery (Lun et al., 2020)

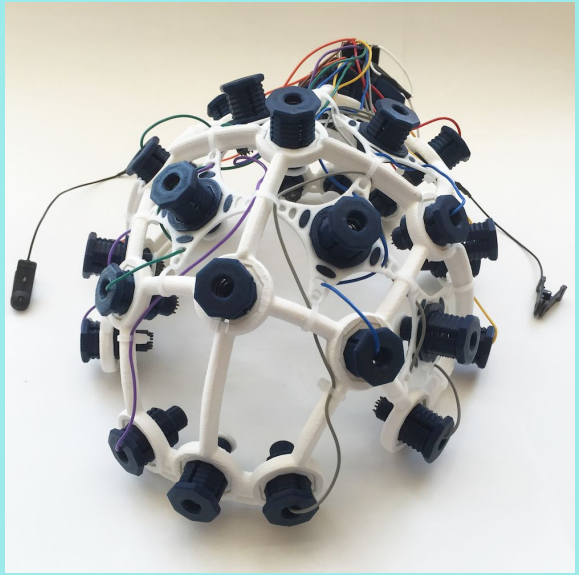
Notes:

- Although our model focuses primarily on hand motion detection, for our bionic hand, our metrics are competitive with results from many other models
- The difference between training and testing metrics show signs of overfitting, more data could help the model generalize better

```
Binary Crossentropy Loss on Training Data: 0.4048079252243042
AUC on Training Data: 0.884029746055603
Classification Accuracy on Training Data: 80%
Precision on Training Data: 88%
Recall on Training Data: 81%

Binary Crossentropy Loss on Test Data: 0.5120210647583008
AUC on Test Data: 0.8099972605705261
Classification Accuracy on Test Data: 74%
Precision on Test Data: 80%
Recall on Test Data: 78%
```

Software/Hardware Integration

Raspberry Pi & OpenBCI

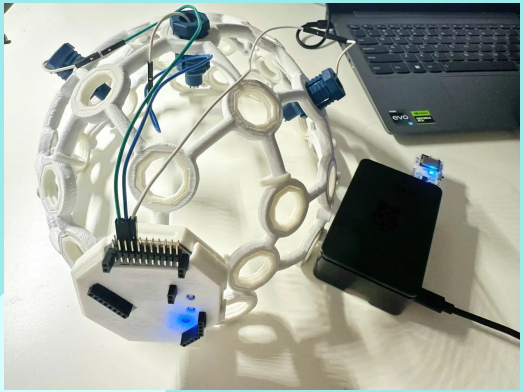


OpenBCI Ultracortex "Mark IV" EEG Headset: A wearable device that captures brain activity (EEG signals) non-invasively, allowing for real-time brain monitoring and interaction with external devices.

Raspberry Pi: A compact and powerful computer that processes the EEG data from the Ultracortex headset, facilitating real-time signal analysis and subsequent robotic control.

Integration Steps

- **Setup and Connection:** Connect the headset to the OpenBCI Cyton board.
- **Data Transmission**: Communicate with Raspberry Pi through OpenBCI dongle.
- **Data Streaming**: Implement BrainFlow to stream EEG data live.
- **Data Processing (previously introduced):** Run signal processing algorithms to clean and interpret the EEG signals.
- **Machine Learning Model Execution (previously introduced)**: Execute the machine learning model to classify EEG signals.



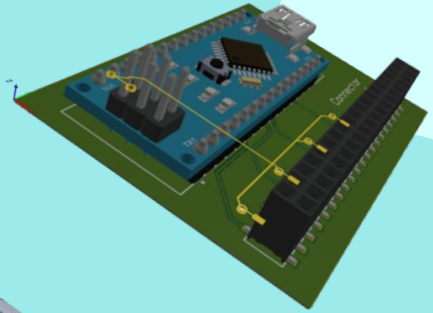
```python
def get_data():
    BoardShim.enable_dev_board_logger()

    parser = argparse.ArgumentParser()
    # use docs to check which parameters are required for specific board, e.g. for Cyton - set serial port
    parser.add_argument('--timeout', type=int, help='timeout for device discovery or connection', required=False,
                        default=0)
    parser.add_argument('--ip-port', type=int, help='ip port', required=False, default=0)
    parser.add_argument('--ip-protocol', type=int, help='ip protocol, check IpProtocolType enum', required=False,
                        default=0)
    parser.add_argument('--ip-address', type=str, help='ip address', required=False, default='')
    parser.add_argument('--serial-port', type=str, help='serial port', required=False, default='')
    parser.add_argument('--mac-address', type=str, help='mac address', required=False, default='')
    parser.add_argument('--other-info', type=str, help='other info', required=False, default='')
    parser.add_argument('--serial-number', type=str, help='serial number', required=False, default='')
    parser.add_argument('--board-id', type=int, help='board id, check docs to get a list of supported boards',
                        required=True)
    parser.add_argument('--file', type=str, help='file', required=False, default='')
    parser.add_argument('--master-board', type=int, help='master board id for streaming and playback boards',
                        required=False, default=BoardIds.NO_BOARD)
    args = parser.parse_args()

    params = BrainFlowInputParams()
    params.ip_port = args.ip_port
    params.serial_port = args.serial_port
    params.mac_address = args.mac_address
    params.other_info = args.other_info
    params.serial_number = args.serial_number
    params.ip_address = args.ip_address
    params.ip_protocol = args.ip_protocol
    params.timeout = args.timeout
    params.file = args.file
    params.master_board = args.master_board

    board = BoardShim(args.board_id, params)
    board.prepare_session()
    board.start_stream()
    time.sleep(10)
    # data = board.get_current_board_data (256) # get latest 256 packages or less, doesnt remove them from internal buffer
    data = board.get_board_data()  # get all data and remove it from internal buffer
    board.stop_stream()
    board.release_session()

    return data
```

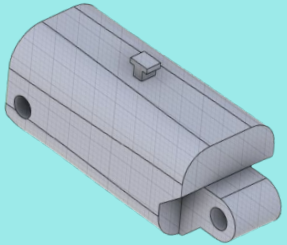
Demo Video

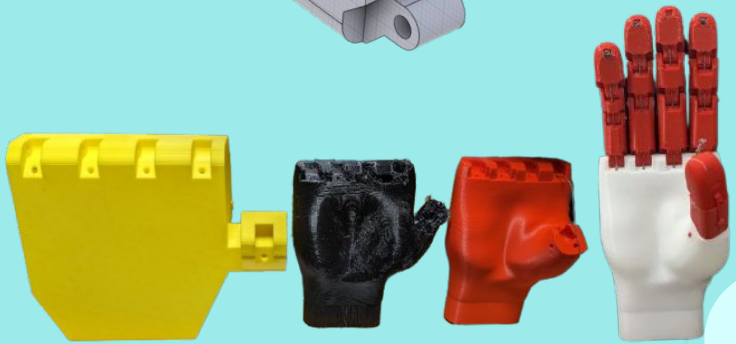
Challenges Faced

- Finding a suitable model architecture
- Determining when our data was denoised
- Finding distinguishable features in the data

Hardware subsystem







Project Timeline

Brainstorming

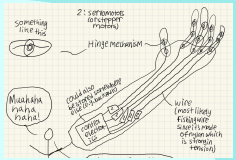
2023 Dec - 2024 Jan

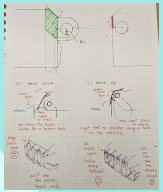
Inspired by other robotic arms such as **Unlimited Tomorrow**

Use a **metal string** to bend the fingers

Use **elastic bands** to straighten them

Design **PCB** for the motor control circuit





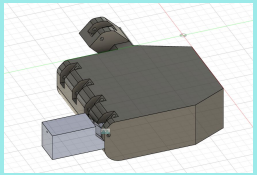
Prototyping

2024 Jan - 2024 Feb

Built prototypes using styrofoam

Built a rough CAD design to confirm that our joint design works

Too bulky and looks awkward



CAD design for ergonomic hand

2024 Feb - 2024 May

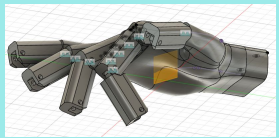
Improved **Ergonomics**

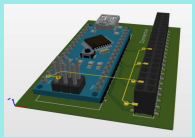
Modelled to **closely resemble** human hands

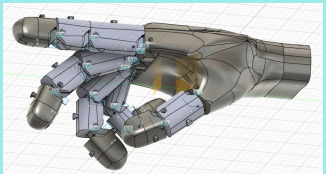
Can **store the motors inside** the palm

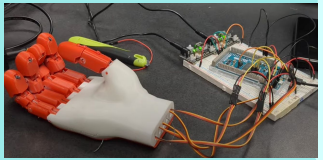
Thumb angle is optimized for **grabbing objects**

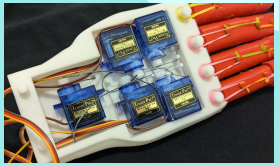
Used **Fusion 360** and **Solidworks** to design on the industry level











Assembly & Integration

2024 May - 2024 Jun

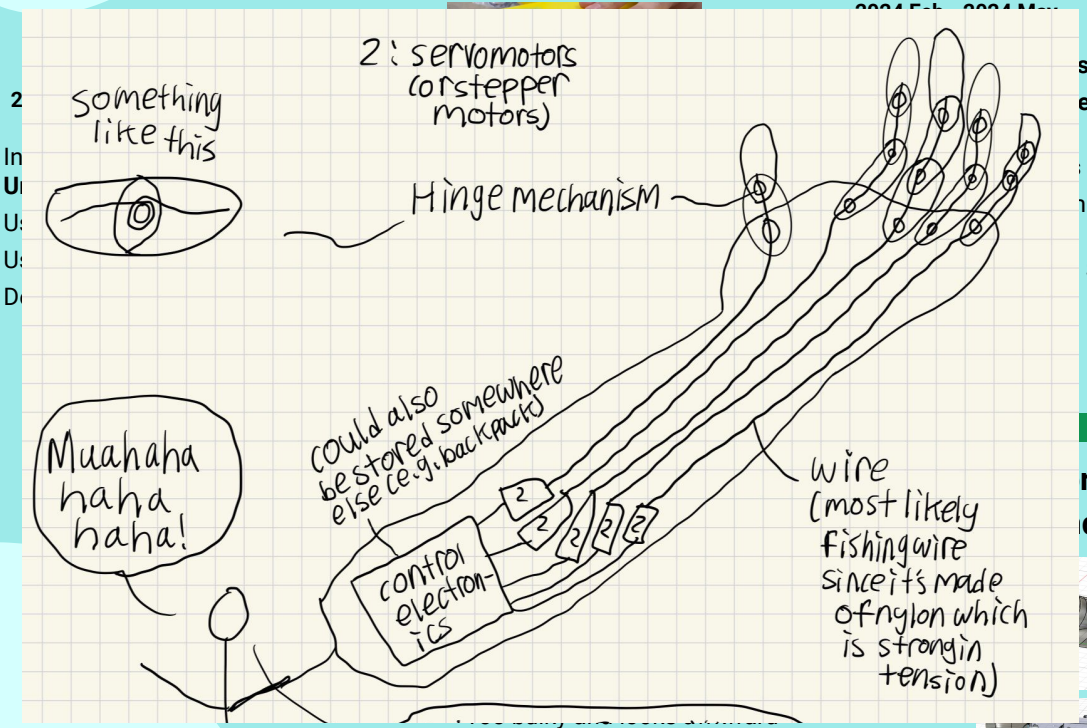
Install the software on a **Raspberry Pi**

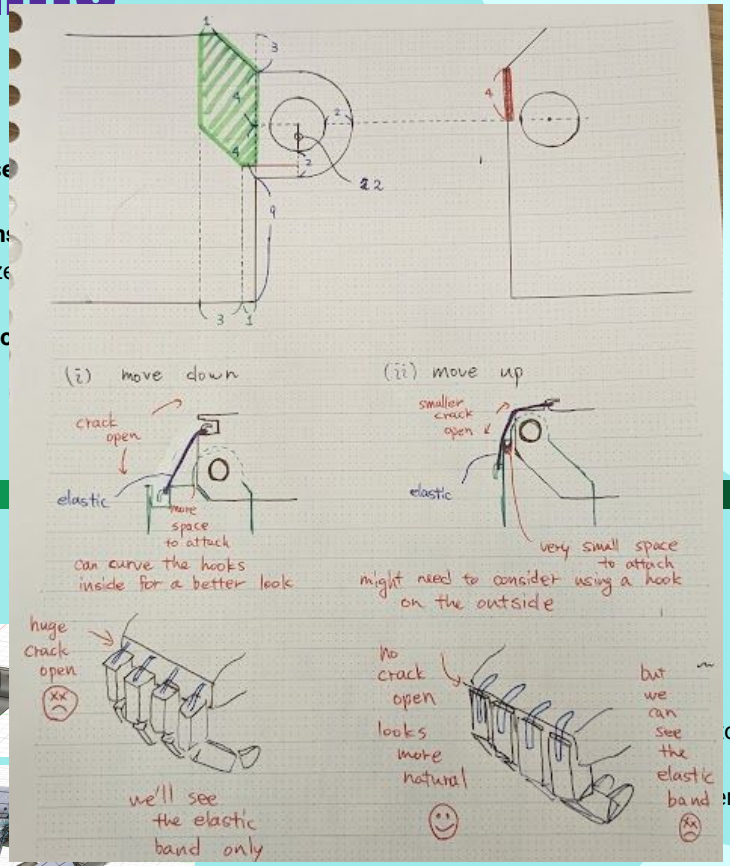
Use **Arduino Due** to control the motors

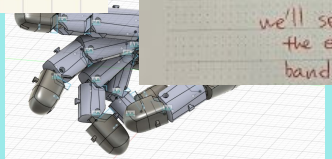
Open BCI headset sends EEG recordings to Raspberry Pi

Raspberry Pi signals Arduino Due to trigger the grasping motion

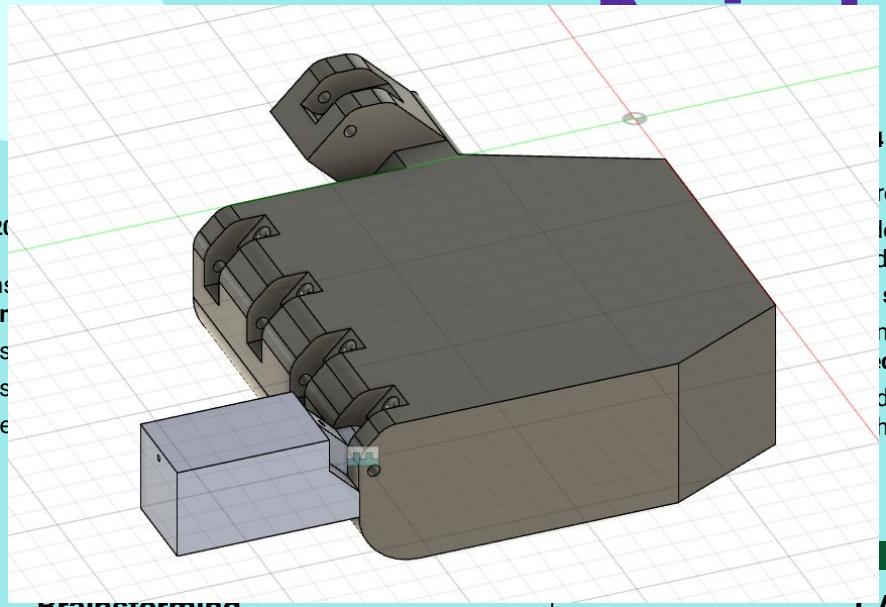
Project Timeline

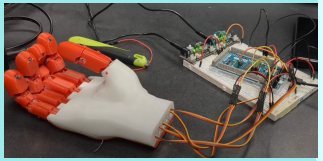


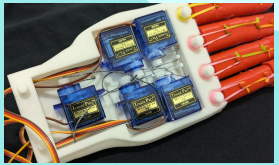




Timeline







Feb - 2024 May

roved **Ergonomics**

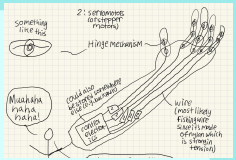
lelled to **closely resemble** human
ds

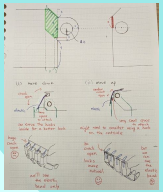
store the motors inside the palm

mb angle is optimized for **grabbing**
cts

d **Fusion 360** and **Solidworks** to design
he industry level

Brainstorming





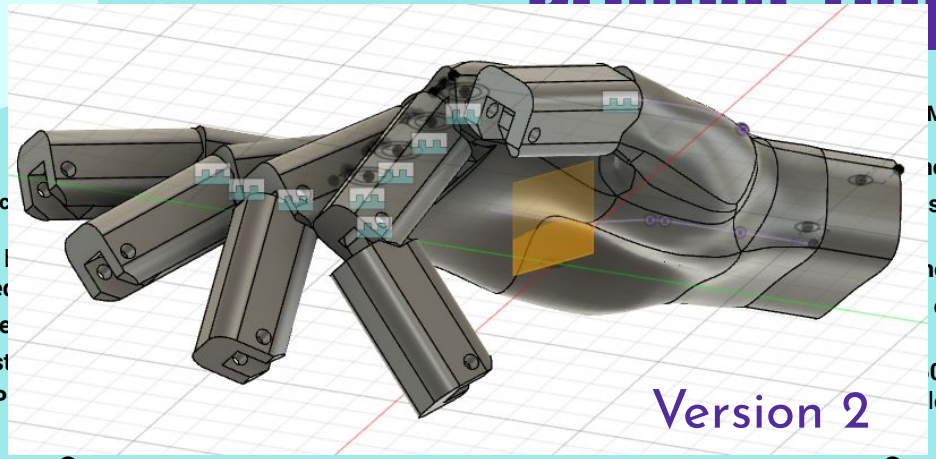
2024 Jan - 2024 Feb

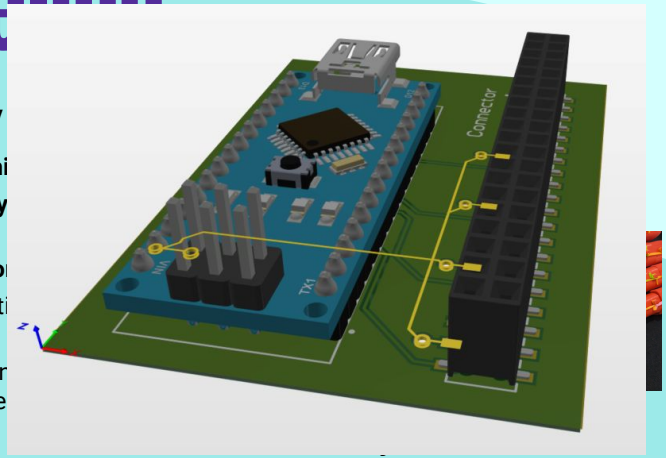
Built prototypes using styrofoam

Built a rough CAD design to confirm that our joint design works

Too bulky and looks awkward

Project Timeline





2023 Dec

Inspired
Unlimite
Use a **me**
Use **elast**
Design **P**

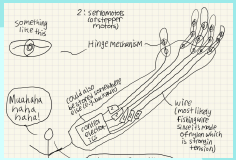
May

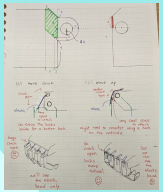
omi
sely
oto
opti
0 ar
leve

Prototyping

Integration

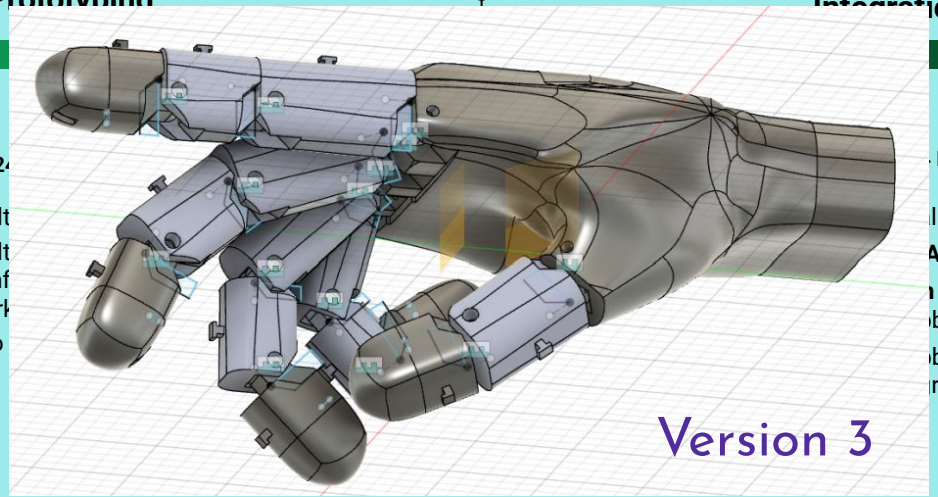
Brainstorming





202

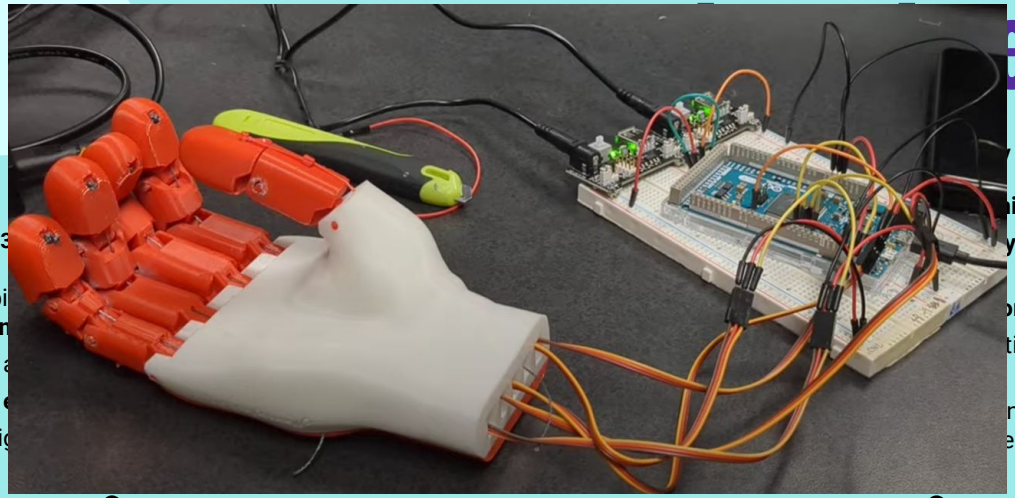
Built
Built
conf
work
Too



May - 2024 Jun

ll the software on a **Raspberry Pi**
Arduino Due to control the motors
BCI headset sends EEG recordings to
berry Pi
berry Pi signals Arduino Due to trigger
rasping motion

eline



2023

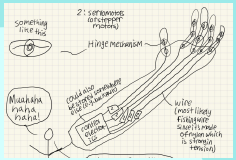
Inspi
Unlin
Use a
Use
Desig

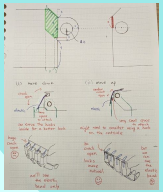
ics
y resemble human

ors inside the palm
timized for **grabbing**

nd **Solidworks** to design
el

Brainstorming





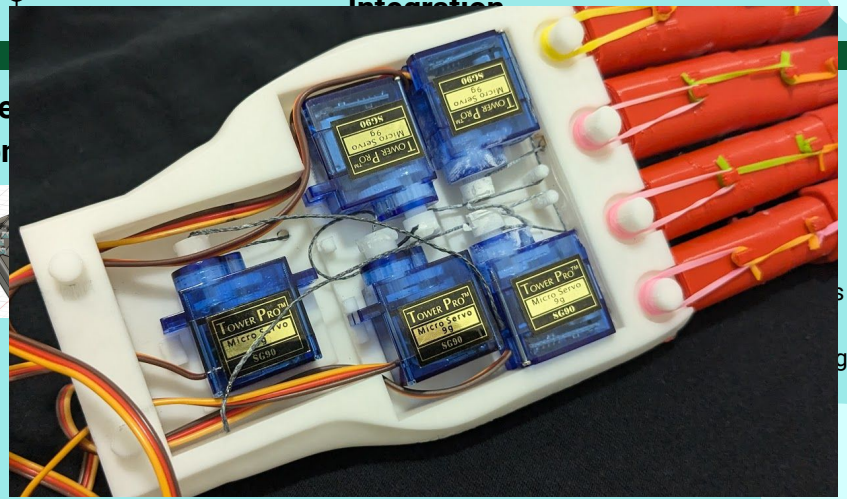
Prototyping

2024 Jan - 2024 Feb

Built prototypes using styrofoam
Built a rough CAD design to confirm that our joint design works
Too bulky and looks awkward

CAD de
ergonor

Assembly &
Integration



s to

ger

Challenges Faced

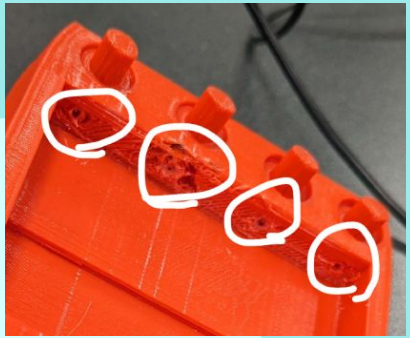
Printing Orientation:

Challenge:

- Printing orientation and deformation from gravity result in poor printing quality.

Solution:

- Used simulation software to test different printing orientations, leading to better print quality.



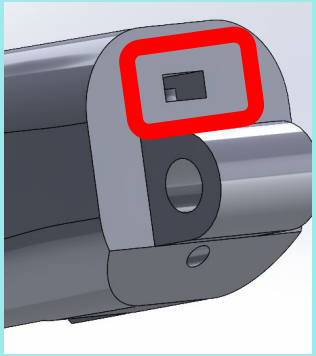
Finger Design:

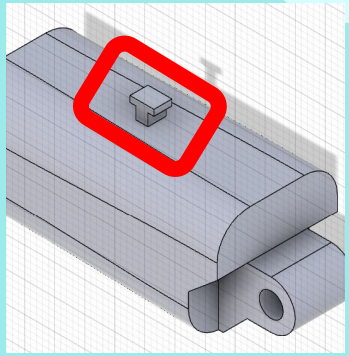
Challenge:

- Rubber bands dislodging during contraction.
- Finger interference issues.

Solution:

- Moved hook for better rubber band grip.
- Added cuts to reduce finger interference.





Challenges Faced - Continued

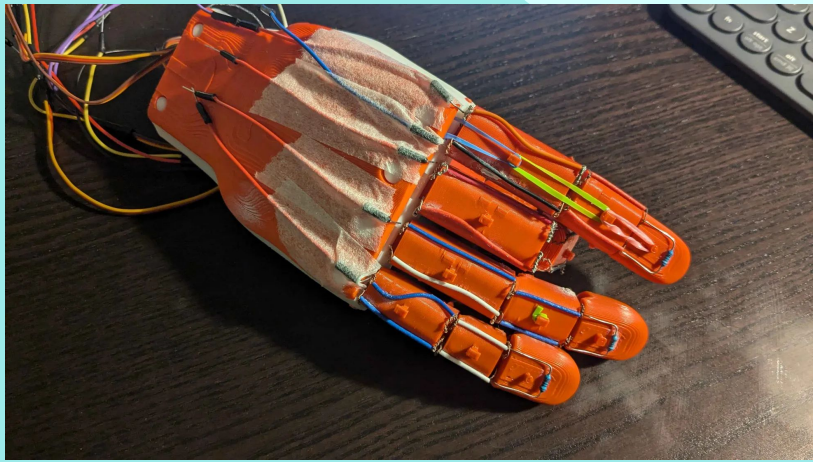
Motor Control:

Challenge:

- A feed forward system provided a limited control over motors, making the system lack adaptability to various scenarios

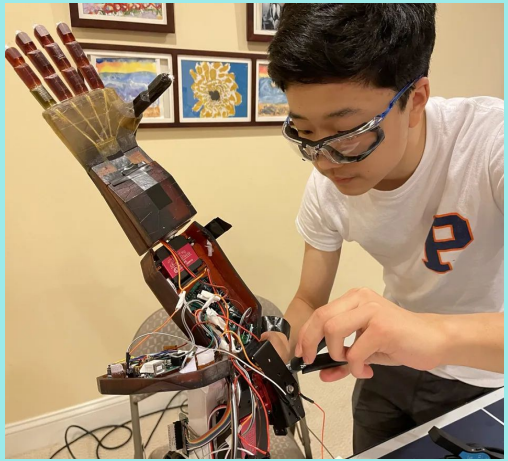
Solution:

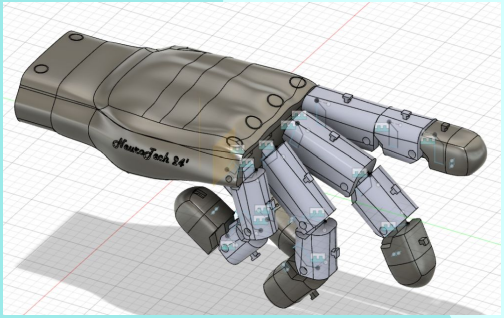
- Implementation of a feedback control mechanism.
  - When the fingers are bent, the circuit opens
  - When the fingers are extended, the circuit closes communicate a signal
  - Microcontroller reads this signal to disable the motors

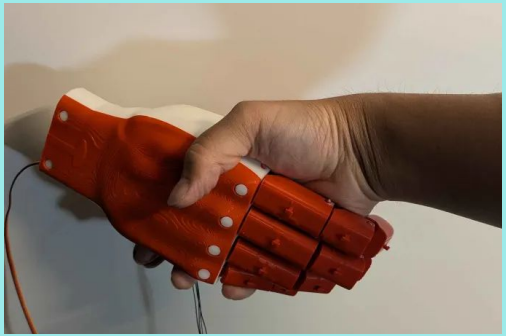


Advantages of Our Prosthetic Arm Design

- Design looks more realistic and is more ergonomic compared to other bulkier prosthetic arm projects out there.
- Optimized space efficiency by housing motors inside the palm, unlike traditional designs that place them in the forearm.
- Suitable for individuals with remaining forearms
- Reduced costs due to leveraging 3D printing for most components, making the design more affordable than many alternatives.







Evaluation of Design

Successes

- Can identify hand opening and closing based on EEG signals
- Prosthetic arm is able to realistically replicate motions of a human hand
- Motion can occur in real time

Limitations

- Can only identify two hand motions
- Accuracy decreases when multiple activities are being performed simultaneously

Future Plans

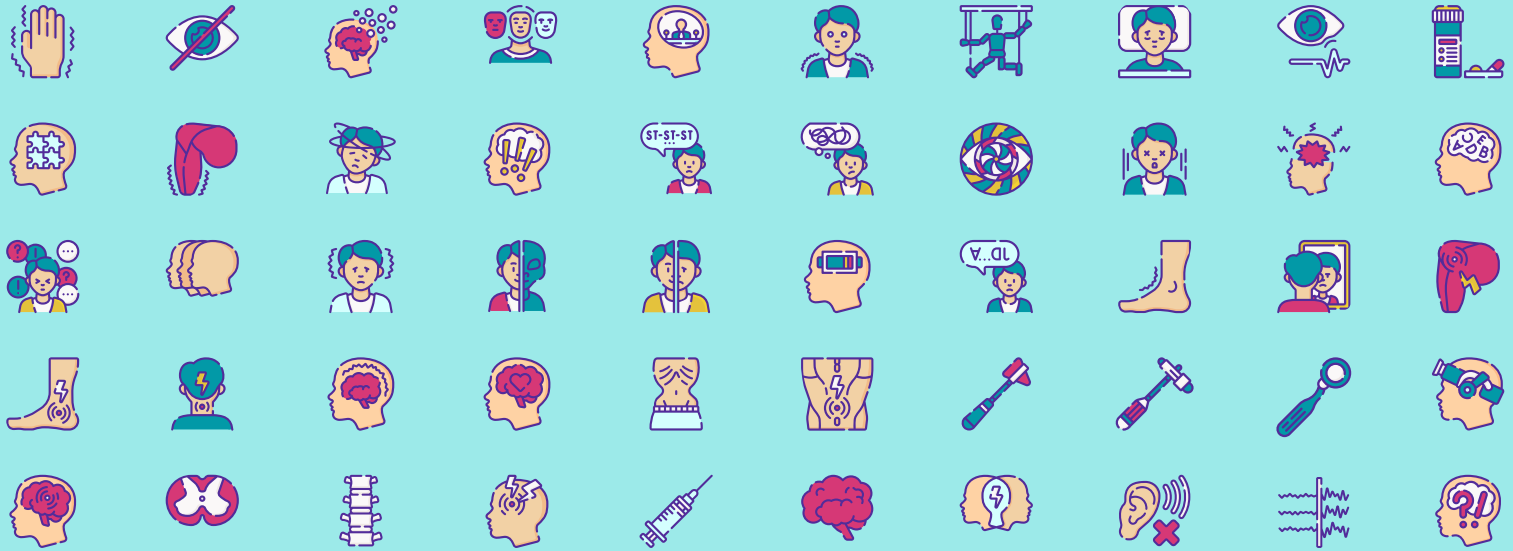
Hardware

- Print the PCB and assemble it for a better portability
- Modify the fingers so that we can replace the rubber bands with more durable elastic component
- Add ball bearings for a smoother finger contraction and reduced loss of torque
- Design a component that attaches the hand to the user's arm with comfort

Software

- Collect and train the model on noisier data, so hand motions will still work in noisy environments, where multiple muscles are moving
- Create models that allow for a larger variety of hand movements to be identified.

ICON PACK



Storyset

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out how it works.

Pana

Amico

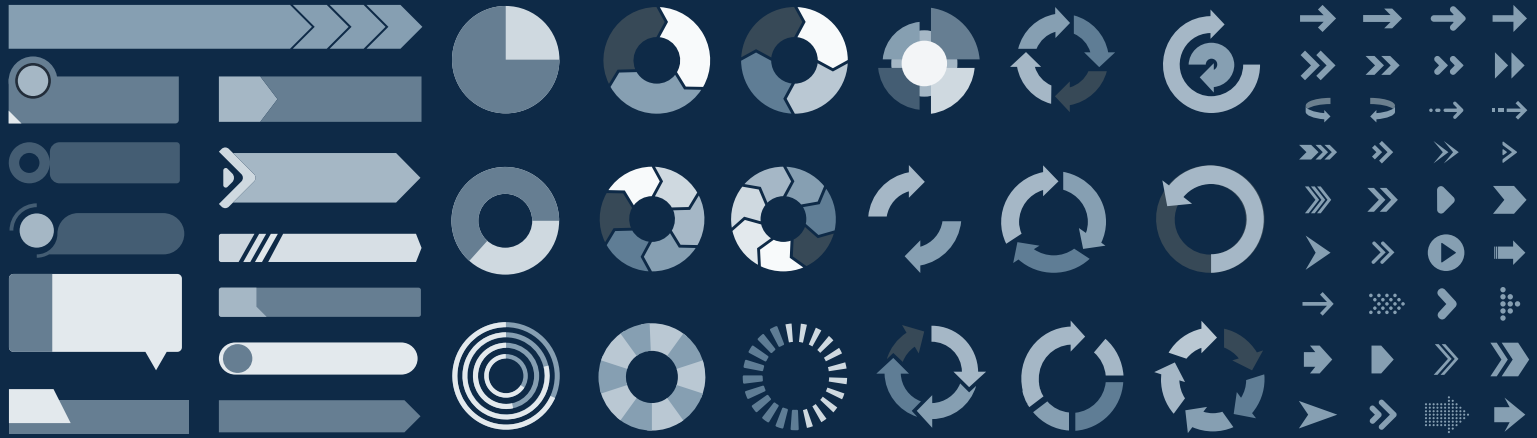
Bro

Rafiki

Cuate

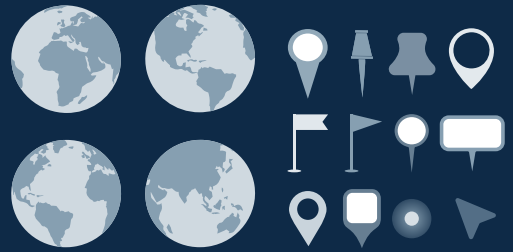
Use our editable graphic resources...

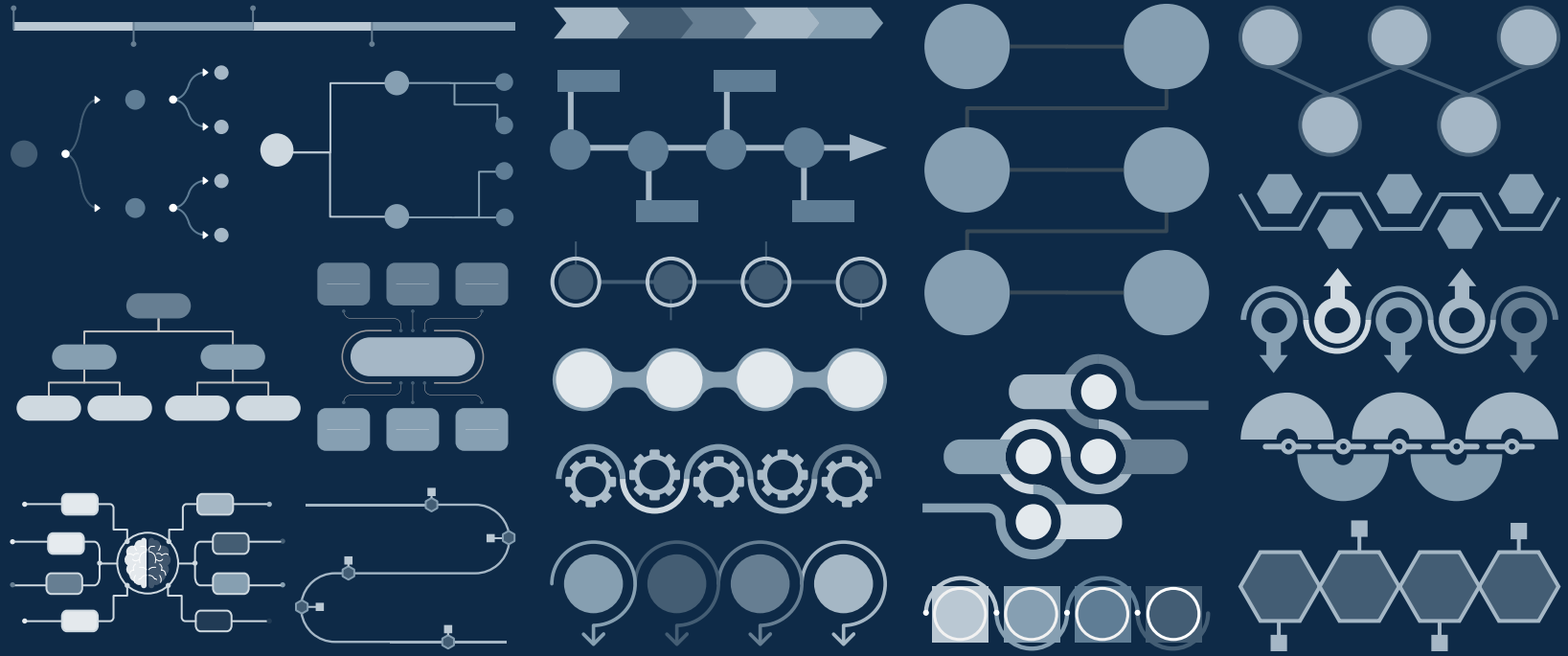
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more infographics on Slidesgo.

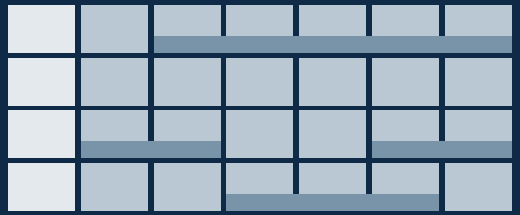


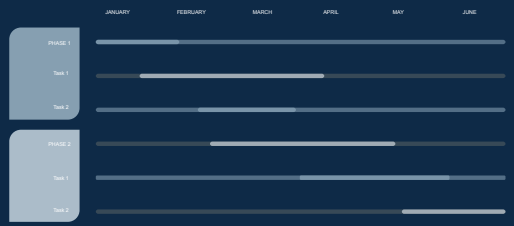


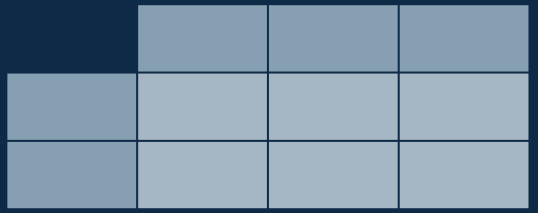


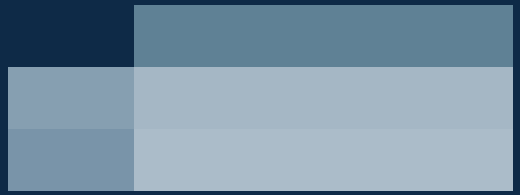



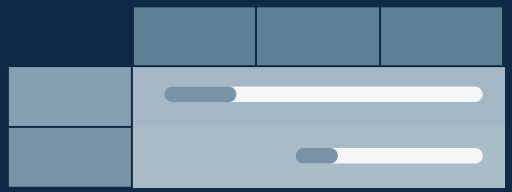


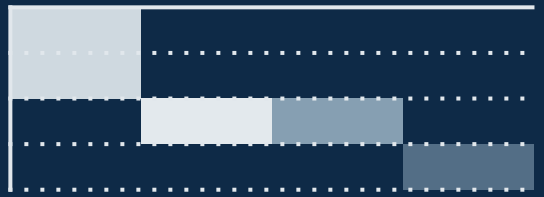


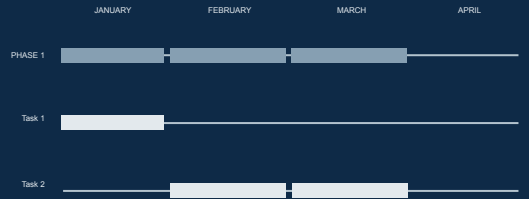

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2

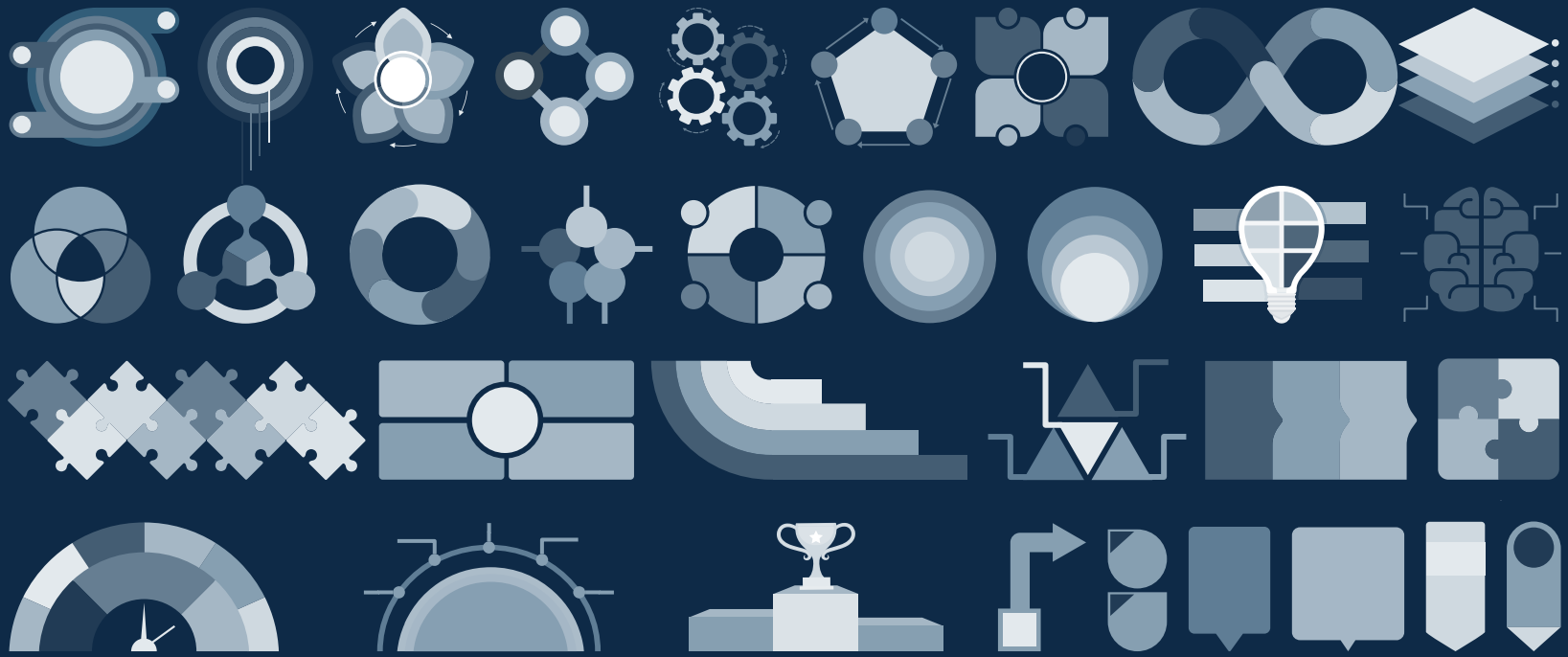









JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2



...and our sets of editable icons

You can resize these icons without losing quality.
You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.
In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons

Help & Support Icons

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo