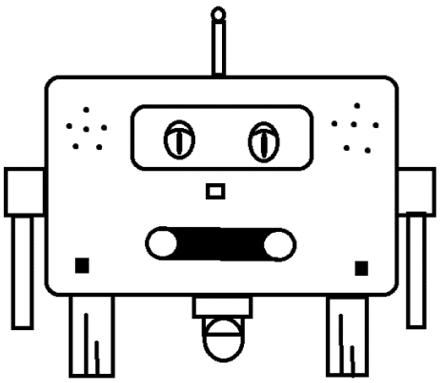


JETSY

Robot asistente personal que interactúa con el ser humano a nivel físico y emocional.

PROJECT SPRINT #2.
DATE: 14th April 2021

Juan Manuel Camara Diaz 1566532
Miguel del Arco Marquez 1566698
Daniel Suárez Valverde 1566103
Christian Ferré Delgado 1565129

Table of Contents

Sprint #1
Date: 14 April 2021

JETSY

Robot asistente personal que interactúa con el ser humano a nivel físico y emocional.

Project description

Este proyecto busca crear un robot autónomo completamente enfocado a una interacción emocional con el usuario. Se llevará a cabo mediante un asistente de mesa con funcionalidades típicas ya existentes pero gracias a la inteligencia artificial podrá interactuar con él mediante la voz y video y así potenciar la interacción robot-humano buscando la máxima fluidez posible.

También creemos que el diseño estético del robot es muy importante para poder transmitir emociones al usuario por lo que estará muy trabajado.

Otro punto que queremos es que todo el software tenga que ser open source, desde los modelos de deep learning hasta las librerías usadas. No necesita conexión a internet para poder utilizar la mayoría de sus funcionalidades.

Electronic components

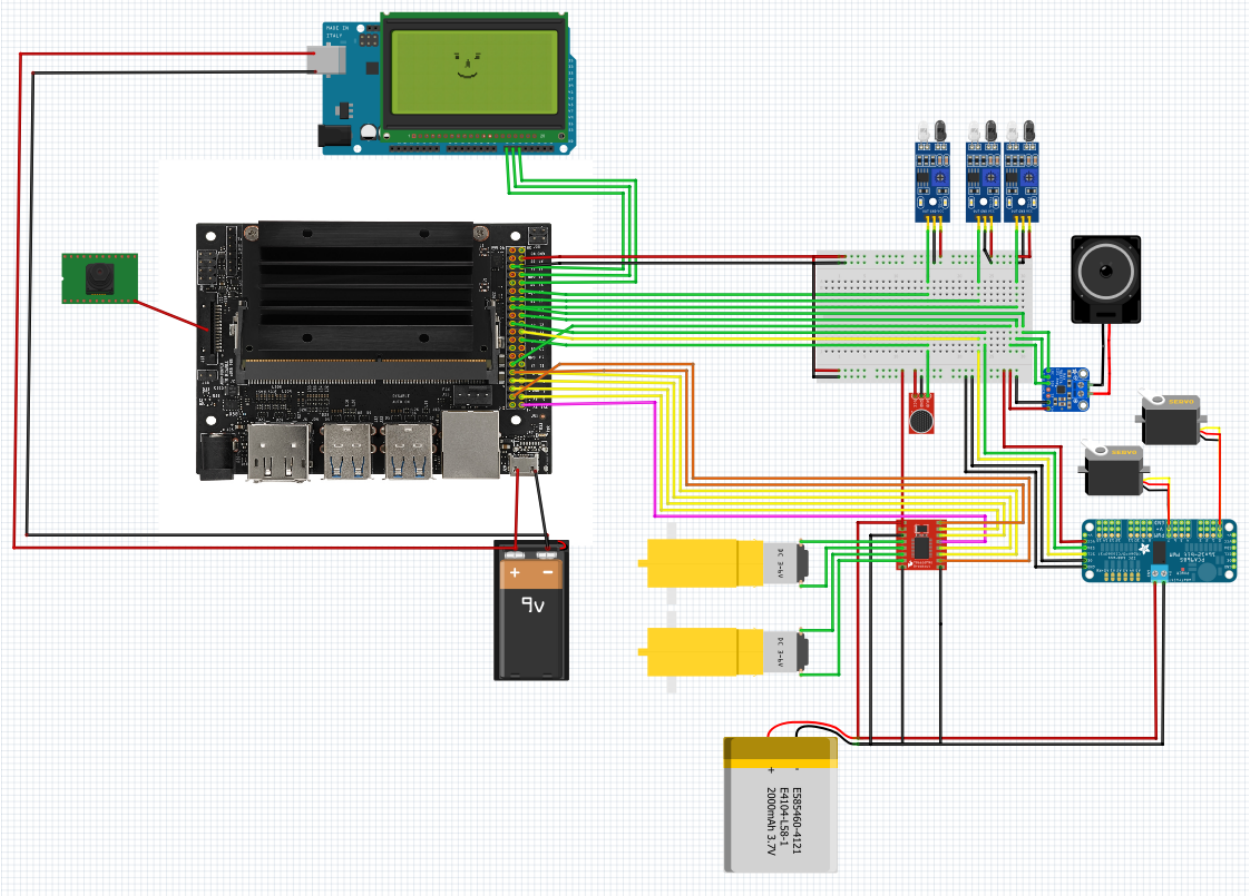
This is the list of the used components:

- Micro servo miniatura SG90 x 3
- Kit motores Dagu 140RPM (2 unidades)
- Controlador de motores TB6612FNG
- Amplificador de audio I2S MAX98357A (3W)
- Batería Lipo 3500mAh 2S 25C, 7.4V

- Arducam 5MP 1080p OV5647 RASPBERRY PI
- Altavoz con caja - 3W
- tft_320qvt_9341
- Micrófono electret preamplificado
- Power Bank 5000 MAh, 1x USB-A
- Raspberry Pi 3 Modelo A+ (Jetson Nano)
- INMSDH32G-100V10
- SENSOR INFRARROJO DE LLAMA x 2
- Controlador de Servos PCA9685

Hardware Scheme

Diagrama del Hardware



- Controlador de Servos PCA9685
  - GND -> GND(38)
  - OE -> GND(38)
  - SCL -> PinOut(10)
  - SDA -> PinOut(32)
  - VCC -> 5V (39)

- Micro servo miniatura SG90 x2
    - VCC -> V+(0,7) (Controlador de Servos PCA9685)
    - GND -> GND(0,7) (Controlador de Servos PCA9685)
    - Pulse -> PWM (0,7) (Controlador de Servos PCA9685)
- Kit motores Dagu 140RPM x2
    - Pin1 -> A01/B02 (Controlador de motores TB6612FNG)
    - Pin1 -> A02/B01 (Controlador de motores TB6612FNG)
- Controlador de motores TB6612FNG
    - VM -> + (Batería Lipo 3500mAh 2S 25C, 7.4V)
    - VCC -> 5V (39)
    - GND -> - (Batería Lipo 3500mAh 2S 25C, 7.4V)
    - PWMA -> PinOut(16)
    - AIN2 -> PinOut(22)
    - AIN1 -> PinOut(18)
    - STBY -> PinOut(21)
    - BIN1 -> PinOut(17)
    - BIN2 -> PinOut(25)
    - PWMB -> PinOut(19)
- Amplificador de audio I2S MAX98357A (3W)
    - VIN -> 5V (39)

- GND -> GND(38)
- DIN -> PinOut(15)
- BCLK -> PinOut(28)
- LRC -> PinOut(8)

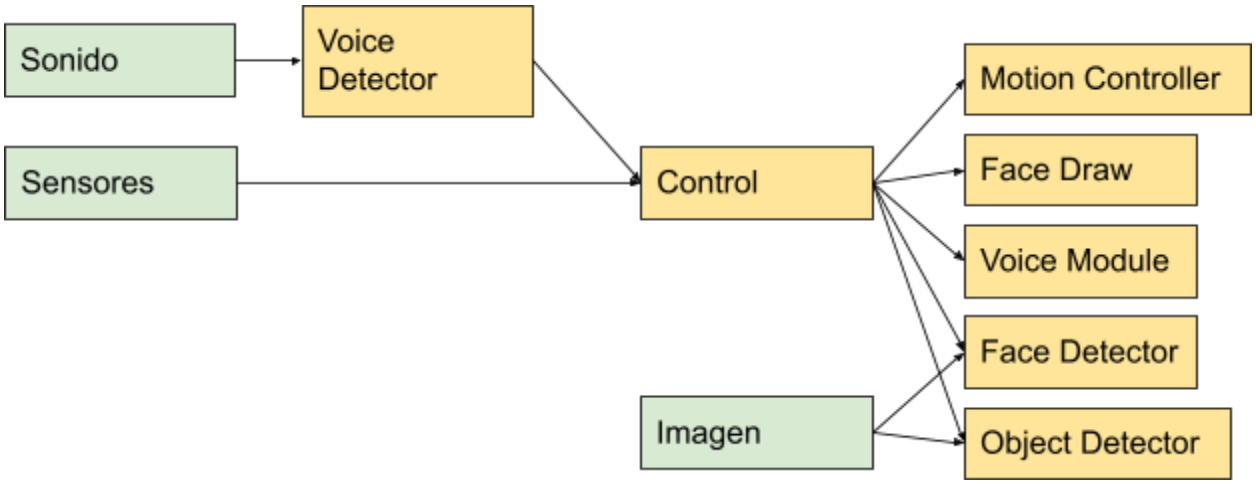
- Batería Lipo 3500mAh 2S 25C, 7.4V
- Arducam 5MP 1080p OV5647 RASPBERRY PI
  - Conexion camara Jetson
- Altavoz con caja - 3W
  - + -> + (Amplificador de audio I2S MAX98357A (3W))
  - - -> - (Amplificador de audio I2S MAX98357A (3W))
- tft_320qvt_9341
  - Conectada directamente al arduino con el adaptador
- Micrófono electret preamplificado
  - VCC -> 5V (39)
  - GND -> GND(38)
  - AUD -> PinOut(29)
- Power Bank 5000 MAh, 1x USB-A
  - Micro USB Jetson
- Raspberry Pi 3 Modelo A+ (Jetson Nano)
- INMSDH32G-100V10

- SD socket

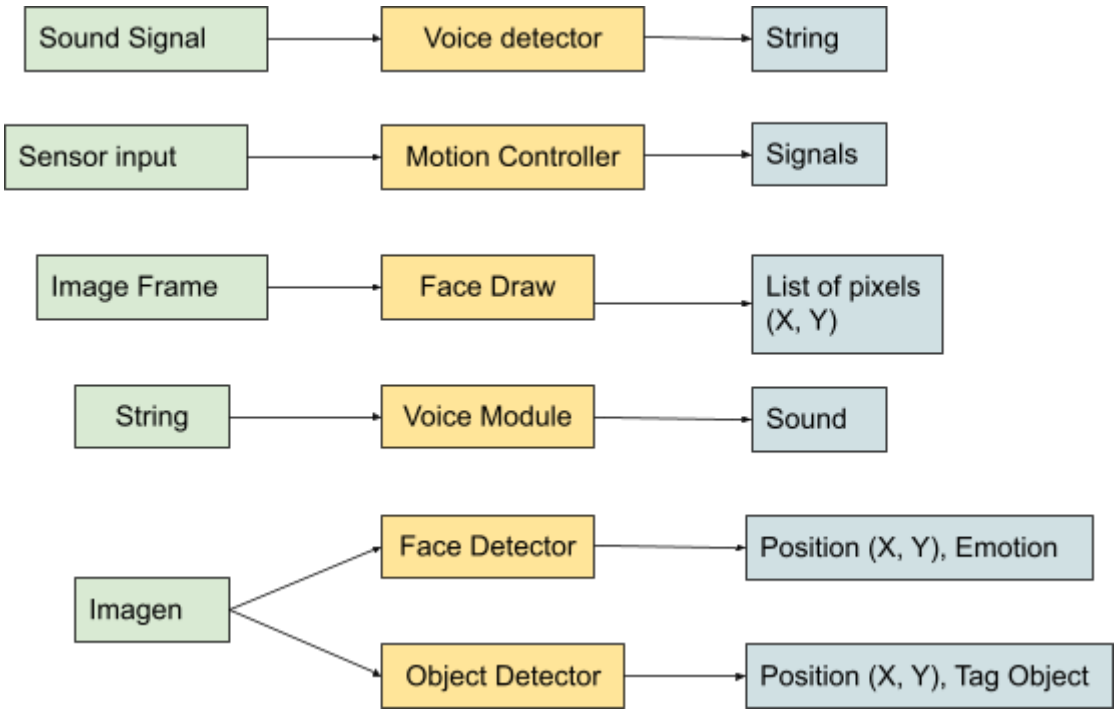
- SENSOR INFRARROJO DE LLAMA x3
  - VCC -> 5V (39)
  - GND -> GND(38)
  - OUT -> PinOut(7, 33, 35)

Software Architecture

Modules diagram



Input-output diagram



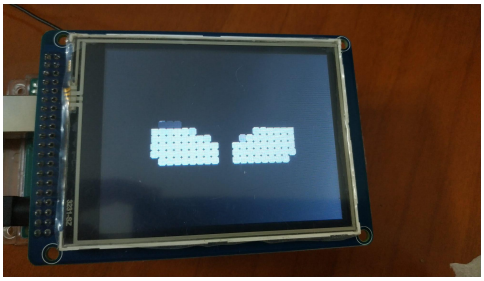
Software modules

Para los diferentes módulos vamos a usar el siguiente software:

Voice Detector: Utiliza un modelo pre-entrenado de reconocimiento de voz, los comandos de voz serán en inglés. El modelo es:
https://github.com/snakers4/silero-models

Monitor Controller: Máquina de estados del robot y quien comunica todos los

Face Draw: Es una máquina de estado de animaciones implementada en el arduino mega. Los estados serán controlados por los pines de input. Para poder almacenar las animaciones de los ojos hemos tenido que crear un script para poder comprimirlas. La compresión se basa en solo almacenar aquellos píxeles que son modificados respecto al frame anterior. Gracias a esto conseguimos comprimir hasta 800 veces de promedio las animaciones. También conseguimos que la animación se vea super fluida aun teniendo una pantalla con una tasa de refresco muy limitada.



Las animaciones comprimidas tienen el siguiente formato (Cada campo es un byte):

[#Frames] [#Cambios frame 0] [#Cambios frame 1] . . . [Cambios frame 0] . . .

Voice Module: Función que recibe un string como input y envía al altavoz la orden para que reproduzca el contenido del string.

La implementación de este módulo ha sido mediante la librería *"pyttsx3"*. Hemos elegido esta librería ya que es offline y muy eficiente.

Face Detector: Función que recibe como input una imagen, ésta es preprocesada para lograr simplificar la información lo máximo posible y, a través de un modelo clasificador devuelve la etiqueta de la emoción que representa la cara.

Utilizará el siguiente modelo de deep learning para detectar la emoción https://github.com/serengil/deepface

Object Detector: Modelo de deep learning encargado de detectar todo tipo de objetos. Es el siguiente: https://pjreddie.com/darknet/yolo/

Amazing contributions

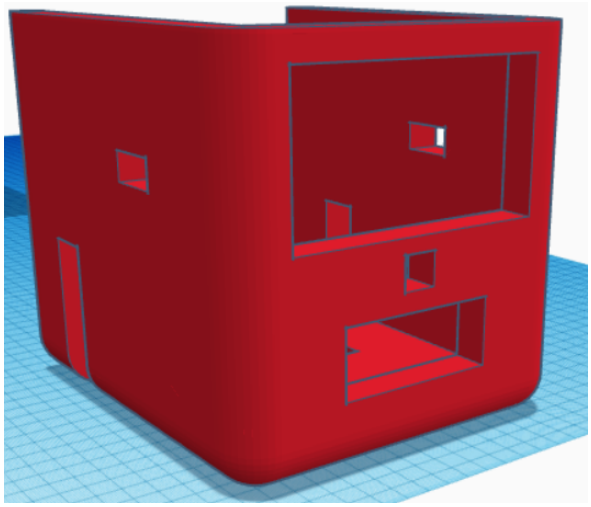
Una interacción humano-robot emocional nunca vista.

Asistente de mesa 2.0 dotado de inteligencia artificial.

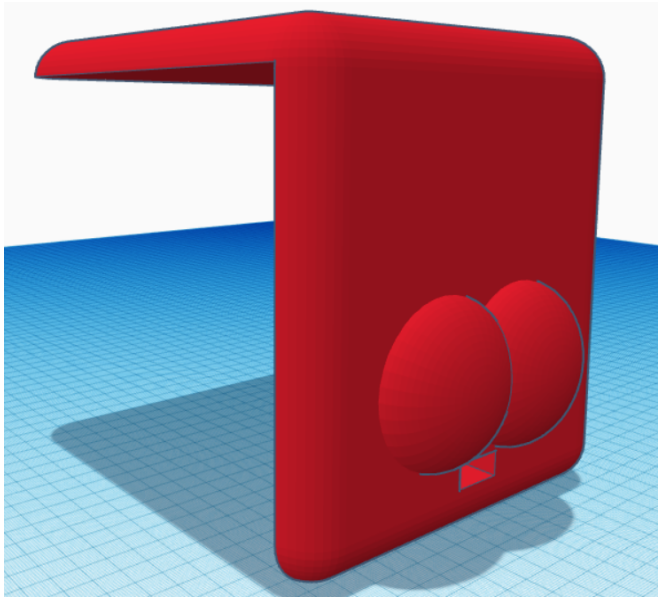
Un asistente con una visión por computador al siguiente nivel. Dejando atrás a los asistentes comerciales actuales.

Extra components and 3D pieces

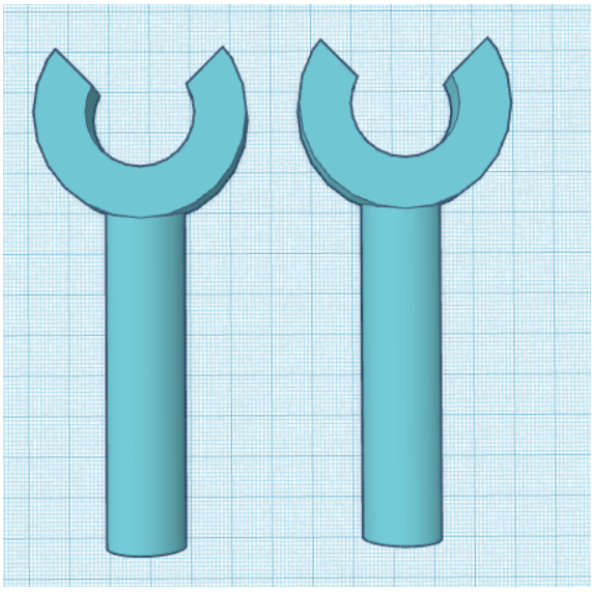
- *Cuerpo del robot (carcasas)*
  - *Frontal*



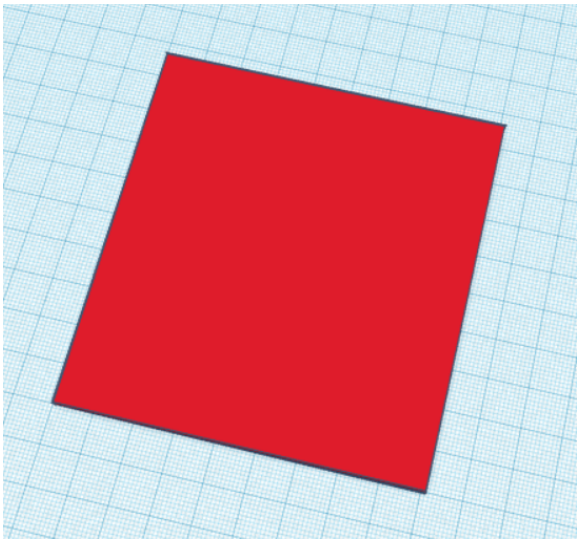
  - *Trasero*



- *Brazo x2*



- *Tapa de pilas*



Simulation Strategy

Para llevar a cabo el robot vamos a crear algunos simuladores mientras se realiza la implementación hardware software.

Para simular una pantalla LCD crearemos un simulador con la librería: PyGame

Al obtener las piezas no hemos optado por simular nada hardware. Todo el software para ser testeado será con un ordenador con Ubuntu 20.04, por lo que no necesitaremos ningún tipo de simulador.

Foreseen risks and contingency plan

Risk#	Description	Probability	Impact	Contingency plan
1	Fallo de Bateria	Low	Low	No usar baterias
2	Bajo rendimiento de CPU/GPU	Low	Low	Reducir el número de módulos de Deep Learning.
3	Máquina de estados compleja	Medium	Low	Reducir las funcionalidades del robot
4	Mal reconocimiento de voz	Low	Medium	Ajustar algunas palabras clave
5	Grandes interferencias de audio	Low	Medium	Reducir la calidad del sonido para evitar las interferencias o usar una api externa.
6	No hacer un proyecto open-source	Medium	Low	Usar código y APIs no open-source
7	Que el robot no se pueda estabilizar bien con las ruedas	Low	Medium	Distribución correcta del peso

8	Falta de espacio dentro del robot	Low	Medium	Hacer un buen diseño 3D de las piezas o hacerlo manual

References

This project has been inspired by the following Internet projects:

- https://living.ai/emo/
- https://developer.amazon.com/es-ES/alexa
- https://www.digitaldreamlabs.com/pages/cozmo

Other links

- https://www.youtube.com/watch?v=8QD6HqL9Qc0&t=7s
- https://pjreddie.com/darknet/yolo/
- https://github.com/snakers4/silero-models
- https://github.com/serengil/deepface