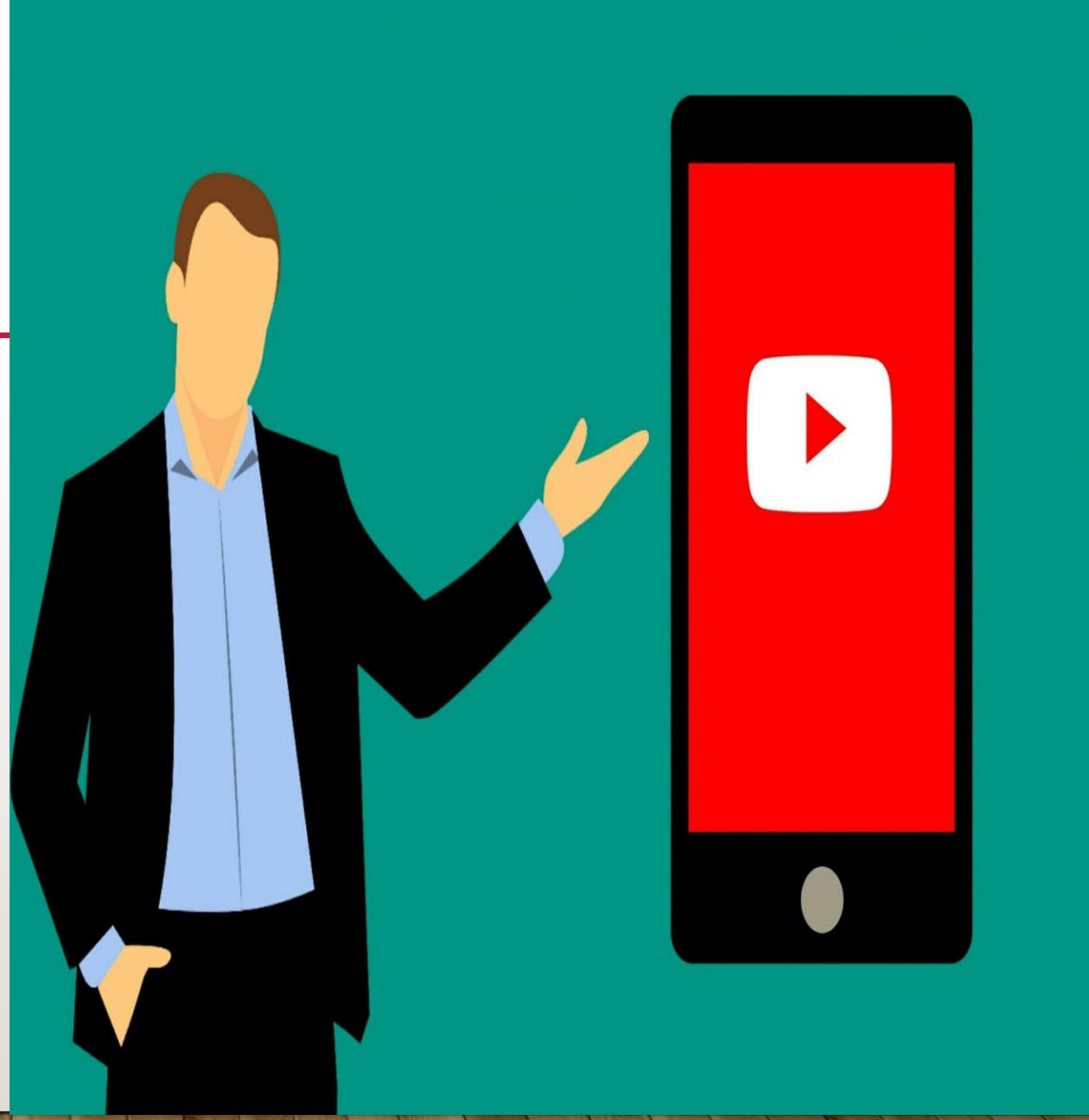
YouTube Video Player

THE IDEA BEHIND THE PROJECT IS FOR THE CLIENT TO MANAGE YOUTUBE FUNCTIONS REMOTELY



SYSTEM

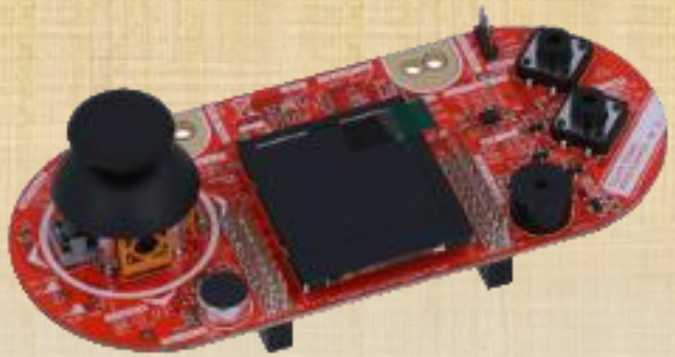
MSP432P401R LAUNCHPAD

development board for the MSP432P401R microcontroller.

ESP32 (ESP32-WROOM-32)

microcontroller with Wi-Fi and Bluetooth for IoT applications.

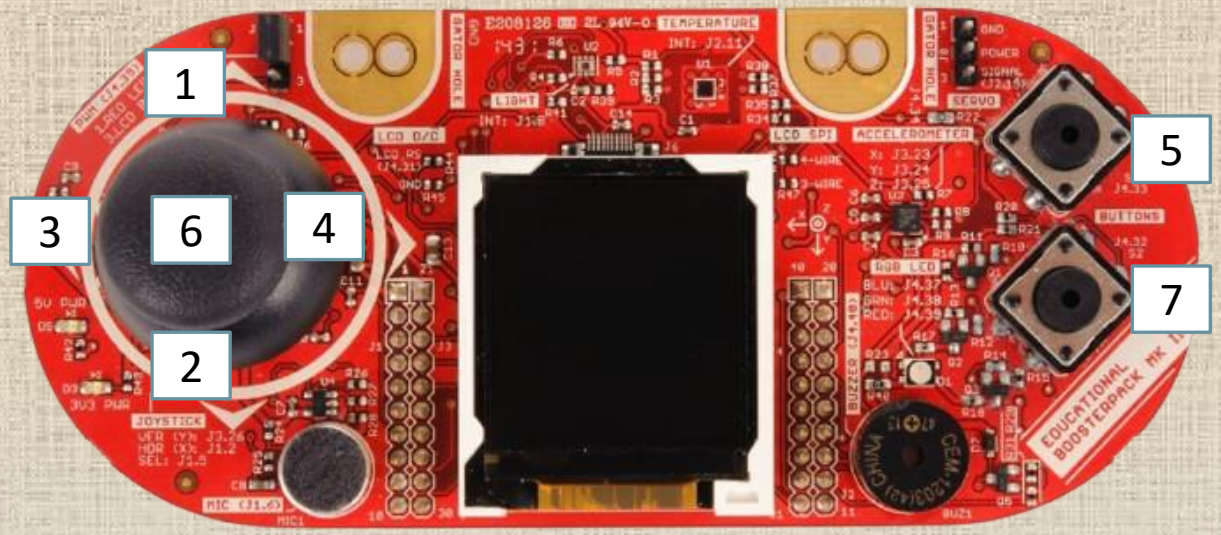
MSP432P401R BOOSTERPACK BOOSTXL-EDUMKII

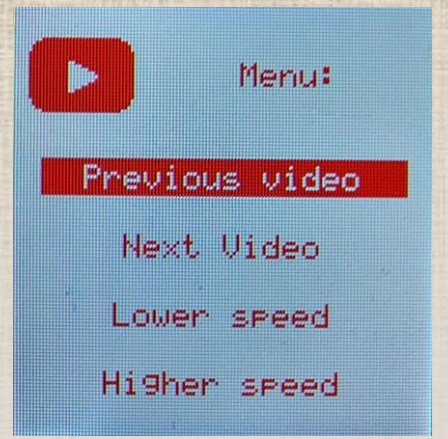


range of sensors and user interface components.

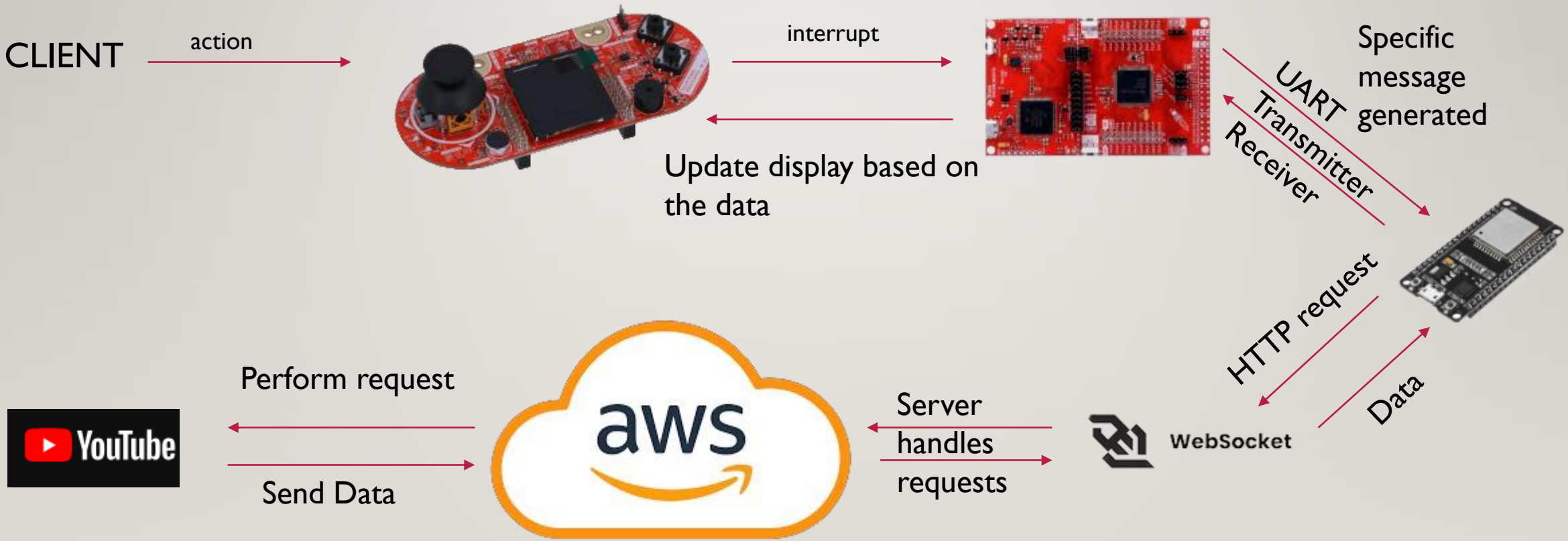
Commands

1. Volume up
2. Volume down
3. 10 seconds backward
4. 10 seconds forward
5. Mute/Unmute
6. Pause/play
7. Enter/exit the menu
    1. Up
    2. Down
    6. Select





WORKING FLOW



CONTROL THE MOVEMENT OF THE JOYSTICK

```
void ADC14_IRQHandler()
{
    uint64_t status;

    status = ADC14_getEnabledInterruptStatus();
    ADC14_clearInterruptFlag(status);

    /* ADC_MEM1 conversion completed */
    if (status & ADC_INT1)
    {
        /* Store ADC14 conversion results */
        resultsBuffer[0] = ADC14_getResult(ADC_MEM0);
        resultsBuffer[1] = ADC14_getResult(ADC_MEM1);

        /* compare the conversion results with given values to
         * enable the interrupt */
        if (resultsBuffer[0] > 15000)
        {
            if (timeDelay && menuOpen == 0)
            {
                if (time <= timeMax - 10)
                {
                    time += 10;
                }
                else
                {
                    time = timeMax;
                    playing = 0;
                    Interrupt_disableInterrupt(INT_TA0_N);
                    _graphics();
                }

                timeDelay = 0;
                char str[10] = "timePlus.";
                sendUART(str);
                _graphics();
            }
        }
        else if (resultsBuffer[0] < 1000)
        {
            if (timeDelay && menuOpen == 0)
            {
                if (time >= 10)
                {
                    time -= 10;
                }
                else
                {
                    time = 0;
                }

                timeDelay = 0;
                char str[11] = "timeMinus.";
                sendUART(str);
                _graphics();
            }
        }

        if (resultsBuffer[1] > 15000)
        {
            if (volumeDelay && menuOpen == 0)
            {
                if (mute)
                {
                    mute = 0;
                    volume = volumeMute;
                }

                if (volume <= 95)
                {
                    volume += 5;
                }
                else
                {
                    volume = 100;
                }

                volumeDelay = 0;
                char str[7] = "volUp.";
                sendUART(str);
                _graphics();
            }
            else if (menuOpen && menuSelect > 0 && volumeDelay)
            {
                menuSelect--;
                --volumeDelay;
                _menuGraphics(menuSelect);
            }
        }
        else if (resultsBuffer[1] < 1000)
        {
            if (volumeDelay && menuOpen == 0)
            {
                if (mute)
                {
                    mute = 0;
                    volume = volumeMute;
                }

                if (volume >= 5)
                {
                    volume -= 5;
                }
                else
                {
                    volume = 0;
                }

                volumeDelay = 0;
                char str[9] = "volDown.";
                sendUART(str);
                _graphics();
            }
            else if (menuOpen && menuSelect < 3 && volumeDelay)
            {
                menuSelect++;
                volumeDelay = 0;
                _menuGraphics(menuSelect);
            }
        }
    }
}
```

PROBLEMS ENCOUNTERED

- No APIs
  - YouTube does not have APIs for its operations
- Client server loaded by ESP32 was too heavy

SOLUTIONS TO PROBLEMS

- Solution to no APIs
  - Communication with WebSockets:
    - 2-way interactive communication session between the browser and the server.
- Set up AWS server to handle the requests

CONCLUSIONS AND FUTURE WORK

It was a great experience to learn how to interact with softwares and sensors.

We also learned how to use WebSocket APIs instead of relying on APIs offered by the server

We were able to set up a server to handle requests

We could improve the system by adding many different streaming platforms on a single page and control them all simultaneously and add different commands