

# PROJECT JOYSTICK

## END USER MANUAL

### **Project Description:-**

The IoT device ESP32(which is running an atSign) will be sensing inputs from a joystick (buttons and the joystick module).

Sending this critical data to the JAVA app (which is running an atSign) so that this data can be sent securely over the Internet.

### **Project Hardware requirements:-**

ESP32,Joystick module with x and y analog outputs and button output,Wiring to connect the joystick module to the ESP32

### **Project Software requirements:**

Java Development Kit (JDK),VScode,PlatformIO

### **Project folders to extract:-**

CS410/Project joystick,JAVA

## **1)CS410/Project joystick documentation**

For sending Joystick data from ESP32

### **Installation:**

-Install **VSCode** and the **PlatformIO** VSCode extension.

-You will need an ESP32 and a USB-A to micro-USB data cable to connect it to your computer.

-The **atPlatform** only supports the Arduino framework for the ESP32.

-Open VSCode and Go to the PlatformIO Home

-Install the **Espressif32** platform by clicking on the **Platforms > Embedded tab** and searching for Espressif32 and clicking on the Install button.

-ESP32 image can be found at **ESP32.HEIC**

## Building the Project:

-Go to Projects and press + Create New Project.

-Give a name to your project, select the **Espressif ESP32 Dev Module board**, and select the "**Arduino**" framework. Select a location for your project and click "Finish."

-Go back to the PlatformIO Home and click on the Libraries tab. Search for at\_client by **JeremyTubongbanua** and click on the Install button. This will install the **at\_client** library in your project. Do the same for the **ArduinoJson** library by **Benoit Blanchon**.

## Extract the folders from CS410/Project-5 into your project folder.

In data folder in the root of the project, copy the **.atKeys** file into the data folder.

The **at\_sign** keys can be found in the link <https://my.atsign.com/login> . By creating atsign account, choose the name for your at\_sign keys to create them. Make sure to download the Backup Keys for the atsign keys you have created.

In **constants.h** file in the include folder in project joystick, change the connection credentials as per the machine the connected to ESP32.

**SSID** "\*\*\*\*\*"

**PASSWORD** "\*\*\*\*\*"

## Uploading the Project:

Open the **main.cpp** file in the src folder and

-Make sure you change the value of joystick pins for input reading according to your joystick connection with ESP32 and make sure you change the name of the sender, recipient **atsign** keys and new at\_key name for the message.

-Run "**Build**" under "Project Tasks". This will give if there are any errors in the project.

-Now upload the project to the ESP32. First, put the ESP32 into download mode by holding down the BOOT button and pressing the RESET button while still holding the BOOT button. It should be in download mode as long as you are holding BOOT down.

-Run "**Upload and Monitor**" under "Project Tasks". Your output should be similar to the snippet below. If it is not, ensure that the .atKeys are uploaded and that you are using the correct SSID and Password.

## OUTPUT SNIPPET:

Attempting to connect to \*\*\*\*\*

..

Connected!

IP address: 12345678901

Connected to root.atsign.org:64

06rcyYJpJkKtcK8eBDGYevE7S+n10E1qM2UJz+m58Uc8VUSR8pCYc6b50x  
mMaQLUjNeQqt2PqyH+gNaKjgQ6UeBdvYpJ01bPawFaITk8k8eu0q6ZGKfbA  
aAuXMDMvwIDAQAB import: 0

complete: 0

public key check: 0

plain\_text: SRKOLuIFV7WKPY1BAaCnRDgFJiUVkvciFXtZSUaBxDw=

73 52 6b 30 6c 75 6c 46 56 37 57 46 50 59 31 42 41 61 43 6e 52 44 67 46 4a  
69 55 56 6b 76 63 69 46 58 74 5a 53 55 61 42 78 44 77 3d

seed: 0

encrypt: 0

command: update:@leopeaceful21:shared key@technological25  
ShF7qKvtNTEGd3zrhceacnLeLeh/qG0K3XMt04PZ1aVIP9VyX9ctQsNZ2mmo  
XXCE32W  
LT+s/B9ZA3cK Yg4RovYacK10qyKYknDOnBbl+YLhA2v+03JWNT  
sm4G/84FSBsAd/aLWzIGD/ME+5WnCifYd5qS9ZGI83PRdvdJF2VBWLCY6J  
3Aks11ljmqPI5e2vQv6lyl3Hjsh1M3FQ/voDfUI1YO fLATNeK3laMRe  
leAqgGNm/Pqptd9HbpFrNCP1CvyKDVx7a/Ck/7IU/tgP90ykdTdkOtPGAvSkT  
a/eSVmCdCbYDdEiYxsaSBurEJZu+5+AUAnPEjS+Np/4BUAA== encrypted  
value: "XhYx7hrVksX+geObiBBJR70TMUgBLaJb3e(D7 r042jU="

command: "up

date: @leopeaceful21: testatechnological25  
XhYx7hrVksX+geObiBBJR70TMUgBLaJb3eD7 r042jU="

**response: "1722"**

When you have receive a value for the response as shown in above snippet, the response value indicates the number of connections made to send data from ESP32, indicating response is success.

After sending the data, delay(100) is called, which pauses the execution of the program for 100 milliseconds before sensing the joystcik values to read and send them.

## 2) JAVA documentation

For Receiving Joystick Data and displaying it in a JAVA SWING(GUI) Application

### Installation:-

- Download and install the **Java Development Kit (JDK)** on your system.
- Download and install an Integrated Development Environment (IDE) for Java development.**VScode** suggested.Install required JAVA extensions.
- Download the JAVA zip file and extract it to a folder
- Open the IDE and create a new project.
- Add the downloaded JAVA files to the project.

### RUN:

- change the name of the sender, recipient atsign keys in App.java.
- make sure the at\_keys in Keys folder inside JAVA are your at\_keys.
- Run the **App.java** file to start the App.

### OUTPUT:

JAVA SWING App!

The console will display the joystick values in textarea of output window,received from the server.

The label in output window shows single point console that moves according to the joystick module movements.

You can find the screenshot of Output Window **OutputScreenshot.png**

X and Y indicates horizontal and vertical movements of joystick.

The X and Y values displayed without any movement in joystick are the default position of console on the Joystick Module and  $z = 1$  indicates that button is pressed.

Move the joystick and look at the X and Y values received. You can also notice console point movements.

Press the **Button** on the joystick to change the color of the black console point to red.

The code continues to execute in an infinite loop, repeatedly fetching data from the shared key, formatting it, and displaying it in the application's UI until the program is terminated externally

You can find a quick explanation video to run the App at <https://www.youtube.com/watch?v=ng-htl6Ms9I>