BLUETOOTH JOYSTICK

-using ESP32 and Dual Axis XY Joystick

Project Report

EKLAVYA MENTORSHIP PROGRAMME

At

SOCIETY OF ROBOTICS AND AUTOMATION,

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

MUMBAI.

OCTOBER 2021.

ACKNOWLEDGMENT

It was a great experience and learnt a lot working with our super cool passionate seniors.They have put in all efforts to guide us and assist us with all resources ,mentor us with timely updates.We have given all our sincerity to the project and could understand our potential to do some great things in field of technology.We have learnt many new things and hope to create jaw-dropping projects in the near future.We are grateful towards you all and thank you for your mentorship programme.Special thanks to SRA for giving us an opportunity and trusting on us which gave us a belief to work hard and learn many great stuff.

Special Thanks to our Obliging Mentors:

- **GAUTAM AGRAWAL**
- **DHAIRYA SHAH**

TEAM EKLAVYA(Bluetooth Joystick)

KRISHNA NARAYANAN

krishnanarayanan132002@gmail.com

+919082411935

OM SHELADIA

omsheladia10@gmail.com

+918655853634

CONTENTS

1. PROJECT OVERVIEW:

1.1 DESCRIPTION OF USE CASE AND PROJECT:

1.The joystick which we are making is used as game controller which can be used in action games (shooting,football,cricket) on the higher level of modification and specification as per the need of the game.The basic joystick module can be used in simple games for (UP and DOWN) control of the respective game or any 2 axis task.This gives a chance to use thye ability of bluetooth

2.The joystick can also be used for big machinery when it comes to industrial use,where the computed system can be given the respective control of the part of machinery and can be handled seamlessly with the joystick.

1.2 TECHNOLOGY USED:

ESP-IDF:

ESP-IDF is Espressif’s official IoT Development Framework for the ESP32, ESP32-S and ESP32-C series of SoCs. It provides a self-sufficient SDK for any generic application development on these platforms, using programming languages such as C and C++.ESP-IDF supports a large number of software components, including RTOS, peripheral drivers, networking stack, various protocol implementations, and helpers for common application use-cases.

BLUEDROID:

Bluetooth is a wireless technology standard for exchanging data over short distances, with advantages including robustness, low power consumption and low cost. The Bluetooth system can be divided into two different categories: Classic Bluetooth and Bluetooth Low Energy (BLE). ESP32 supports dual-mode Bluetooth, meaning that both Classic Bluetooth and BLE are supported by ESP32.

FreeRTOS :

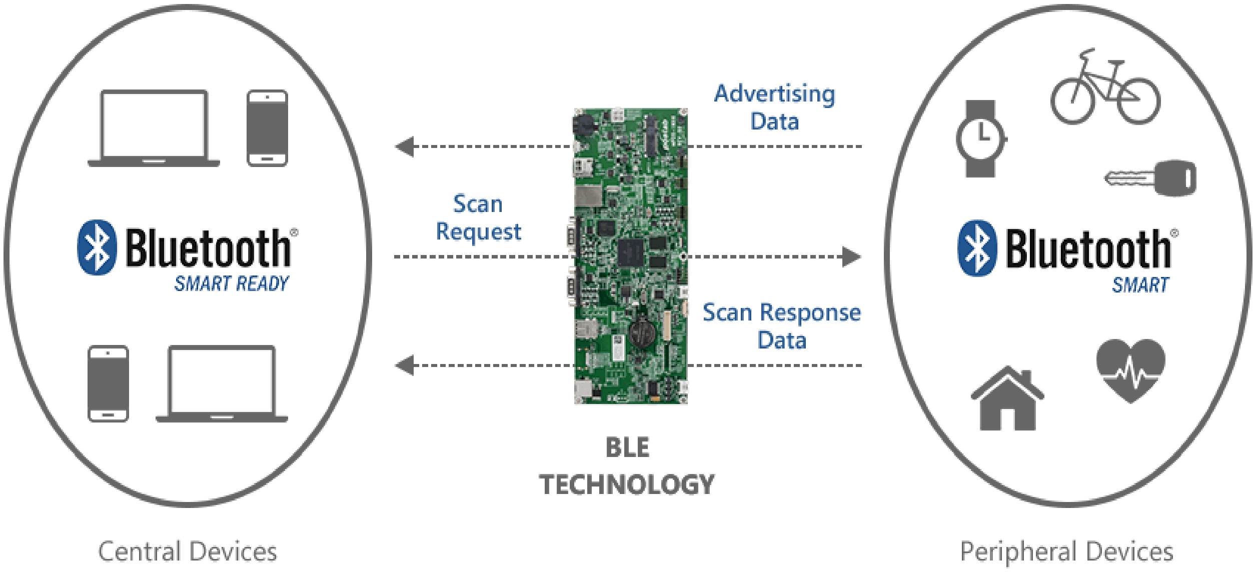
It is an open source, real-time operating system for microcontrollers that makes small, low-power edge devices easy to program, deploy, secure, connect, and manage.

FreeRTOS) is an operating system for microcontrollers that makes small, low-power edge devices easy to program, deploy, secure, connect, and manage. On the other hand, Linux is detailed as "A family of free and open source software operating systems based on the Linux kernel"

FreeRTOS is a class of RTOS that is designed to be small enough to run on a microcontroller - although its use is not limited to microcontroller applications. ... FreeRTOS therefore provides the core real time scheduling functionality, inter-task communication, timing and synchronisation primitives only.

BLE:

Bluetooth Low Energy (BLE) is a low power wireless communication technology that can be used over a short distance to enable smart devices to communicate. ... It has enabled device manufacturers to add a low power communications interface on existing solutions.The Bluetooth Low Energy (LE) radio is designed for very low power operation



The Bluetooth Low Energy (LE) radio is designed for very low power operation. Transmitting data over 40 channels in the 2.4GHz unlicensed ISM frequency band, the Bluetooth LE radio provides developers a tremendous amount of flexibility to build products that meet the unique connectivity requirements of their market. Bluetooth LE supports multiple communication topologies, expanding from point-to-point to broadcast and, most recently, mesh, enabling Bluetooth technology to support the creation of reliable, large-scale device networks.

GATT Services:

A GATT characteristic is a basic data element used to construct a GATT service, BluetoothGattService . The characteristic contains a value as well as additional information and optional GATT descriptors, BluetoothGattDescriptor.GATT is an acronym for the Generic Attribute Profile, and it defines the way that two Bluetooth Low Energy devices transfer data back and forth using concepts called Services and Characteristics.

HID Devices:

A human interface device or HID is a type of computer device usually used by humans that takes input from humans and gives output to humans. The term "HID" most commonly refers to the USB-HID specification.The HID standard was adopted primarily to enable innovation in PC input devices and to simplify the process of installing such devices. Prior to the introduction of the HID concept, devices usually conformed to strictly defined protocols for mouse, keyboards and joysticks.In the HID protocol, there are 2 entities: the "host" and the "device". The device is the entity that directly interacts with a human, such as a keyboard or mouse.

1.3 BRIEF IDEA:

This projects aims to make a joystick which can be controlled using ESP 32 , this device and the host device/ game's brain controller will be connected over bluetooth. The ESP32 is capable of getting a unique signal for each key press. Accordingly it will transmit the signal information over bluetooth to the host device.

2. INTRODUCTION:

2.1 GENERAL:

This project is aimed at making a bluetooth joystick.The ESP32 microcontroller is used as the brain of the the system and it mediates between the joystick and the computer(game or for any usage).As the joystick is moved in a certain direction(2-D movement) it returns values

according to which the game controls can be manipulated.

For this project we have basically used different concepts like HID and GATT.Studying about different protocols used.To analyse the raw readings(ADC values)received to understand the movement of joystick.In this project we have connected the esp32 ,joystick over the bluetooth to the computed system and make it a wireless joystick as soon as we flash and connect it to an external battery source.

2.2 BASIC PROJECT DOMAINS:

- ESP-IDF
- ESP32(Bluetooth version and BLE)
- BLE and Bluedroid
- FreeRtos and GATT Services
- I2C and SPI protocol
- ADC(Diff.channels used)

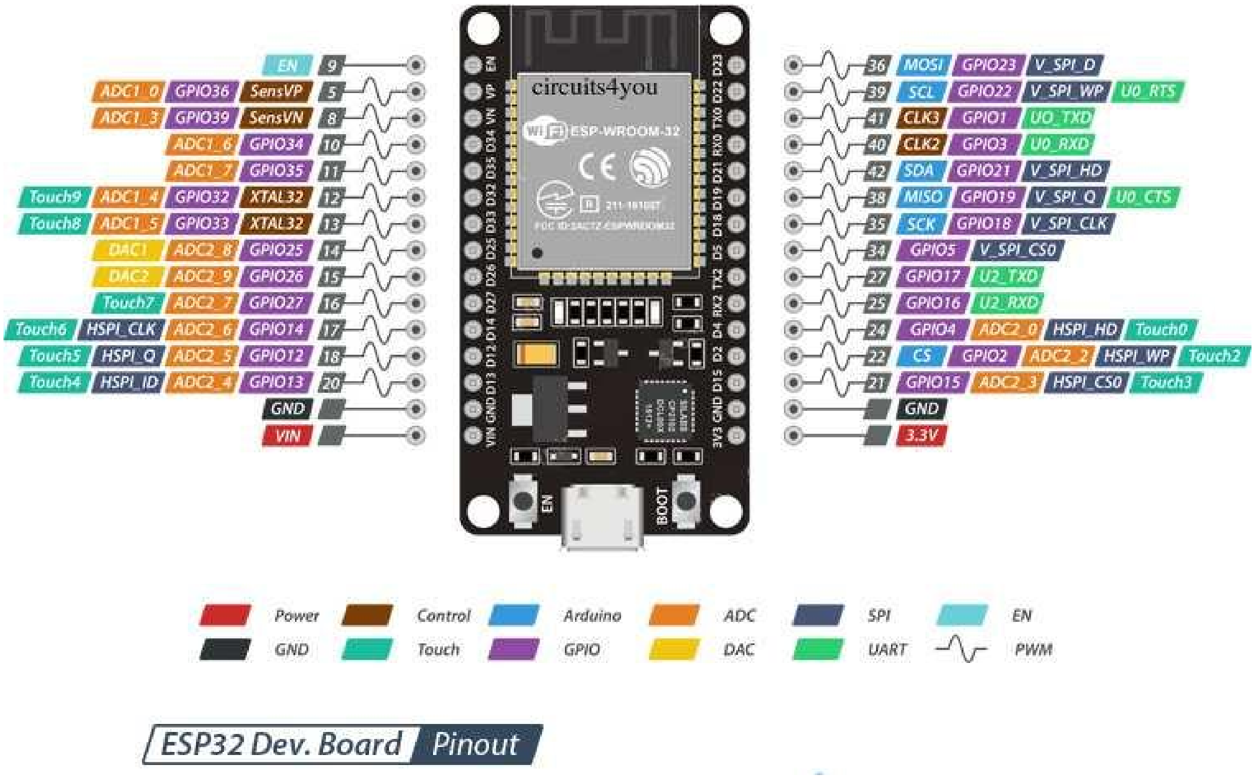
2.3 THEORY :

The main motto of the project is to make the setup wireless.This can be achieved by using the ability of ESP-32 to have BLE within it with which it can be connected over bluetooth to the Computer System and returns the value according to the movement of the joystick.The esp-idf(Espressif's official IoT Development Framework)which was the main resource for our project,all the information about the pinning system and channels present have to be read and their use have to studied from here.

The values received from the ADC are the voltage values(raw values),as voltage values which are analog are first converted to binary and then to digital format,So it is obtained in the raw form.The joystick depending on the requirement of the game or task,its use can be changed and modified in 2-D constraint(joystick supports 2-D movements only).

3.Hardware:

3.1 ESP32


EN 9
ADC1_0 GPIO36 SensVP 5
ADC1_3 GPIO39 SensVN 8
ADC1_6 GPIO34 10
ADC1_7 GPIO35 11
Touch9 ADC1_4 GPIO32 XTAL32 12
Touch8 ADC1_5 GPIO33 XTAL32 13
DAC1 ADC2_8 GPIO25 14
DAC2 ADC2_9 GPIO26 15
Touch7 ADC2_7 GPIO27 16
Touch6 HSPI_CLK ADC2_6 GPIO14 17
Touch5 HSPI_Q ADC2_5 GPIO12 18
Touch4 HSPI_ID ADC2_4 GPIO13 20
GND
VIN
EN VP VN D34 D35 D32 D33 D25 D26 D27 D14 D12 D13 GND VIN
D23 D22 TX0 RX0 D21 D19 D18 D5 TX2 RX2 D4 D2 D15 GND 3V3
circuits4you
ESP-WROOM-32
EN
BOOT
36 MOSI GPIO23 V_SPI_D
39 SCL GPIO22 V_SPI_WP U0_RTS
41 CLK3 GPIO1 U0_TXD
40 CLK2 GPIO3 U0_RXD
42 SDA GPIO21 V_SPI_HD
38 MISO GPIO19 V_SPI_Q U0_CTS
35 SCK GPIO18 V_SPI_CLK
34 GPIO5 V_SPI_CS0
27 GPIO17 U2_TXD
25 GPIO16 U2_RXD
24 GPIO4 ADC2_0 HSPI_HD Touch0
22 CS GPIO2 ADC2_2 HSPI_WP Touch2
21 GPIO15 ADC2_3 HSPI_CS0 Touch3
GND
3.3V
Power
Control
Arduino
ADC
SPI
EN
GND
Touch
GPIO
DAC
UART
PWM
ESP32 Dev. Board Pinout

ESP32 is a system on a chip that integrates the following features:

- Wi-Fi (2.4 GHz band)
- Bluetooth
- Dual high performance Xtensa® 32-bit LX6 CPU cores
- Ultra Low Power co-processor
- Multiple peripherals

> Powered by 40 nm technology, ESP32 provides a robust, highly integrated platform, which helps meet the continuous demands for efficient power usage, compact design, security, high performance, and reliability.

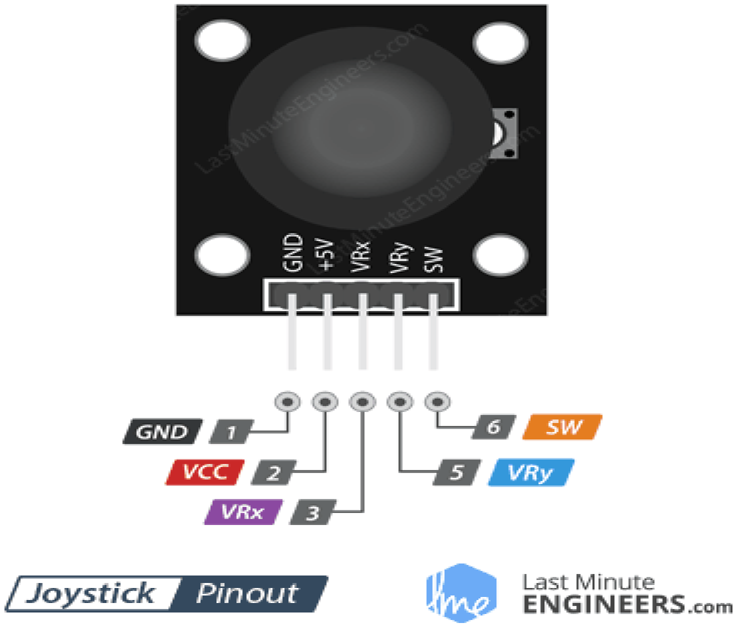
ESP32 is a feature-rich MCU with integrated Wi-Fi and

Bluetooth connectivity for a wide-range

of applications.The ESP32, as a single 2.4 GHz Wi-Fi and Bluetooth combo chip, supports Wi-Fi setups via both SmartConfig and Bluetooth. Users can use ESP32 for secure configuration of Wi- Fi networking for IoT devices. ... ESP32 is compliant with Bluetooth v4. 2 BR/EDR and BLE specifications.

ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces.ESP32 achieves ultra-low power consumption with a combination of several types of proprietary software. ESP32 also includes state-of-the-art features, such as fine-grained clock gating, various power modes and dynamic power scaling.

3.2 DUAL AXIS JOYSTICK:



This is a dual axis high quality JoyStick Module . It can be used to sense movements in 2 directions(axes). It also has a inbuilt switch which can be activated by pressing the stick.Directional movements are simply two potentiometers - one for each axis. Pots are ~10k each. With the help of this Joystick Module, you can measure position coordinates on the X and Y axis by moving the "hat". It also contains a switch that is press-able by pushing the "hat".It also contains a switch that is press-able by pushing the "hat" down. Similar to the XBOX controller.

The X and Y axes are two 10k potentiometers which control 2D movement by generating analog signals. When the module is in working mode, it will output two analog values, representing two directions. This module uses the 5V power supply, and value, when reading through analog input, would be about 2.5V, a value will increase with joystick movement and will go up till maximum 5V; the value will decrease when the joystick is moved in other direction till 0V.

4.METHODS AND STAGES OF PROGRESS:

4.1 APPROACH:

The approach of the project is quite simple and understandable First of all its aim is to make a bluetooth joystick.To take a look at the hardware stuff we need a joystick module(2-D module in this project),a ESP-32 microcontroller(according to the version kitc the pinout may vary),jumper wires for connection.For the bluetooth establishment we tried by connecting our esp32 as a mouse.To understand when the connection is established how does the mouse behave as in the cursor.Next we start to connect the proper pinout connection of the ESP-32 to the Joystick.

So we want the system to know that if a joystick is moved the ESP-32 send or returns some value,for that we have the concept of ADC,this continuously send the instant voltage values and when we move our joystick in different direction(2-D) it returns different values(extreme values are expected).Then finally we have integrate both together that is the bluetooth part and joystick returned values(which will tell the movement according to our code).This will complete our project and can be modified to be used in games.

4.2 Bluetooth Establishment:

For the bluetooth establishment part,HID plays a very important role ,HID Reports.The Bluetooth HID profile allows users to control the HID descriptor, which defines the device's feature set, and the HID report, which the host uses to interpret the data as movement. The host driver software decodes the raw HID report and passes the key values to the application running on the computer system.We are using multiple HID functions to enable the connection.GATT and GAP services are also used for bluetooth enhancement.We are using different types of identities to know or confirm whether the establishment is successful or not ,When we connect it as a mouse and defined its coordinates,the cursor is positioned at that coordinates which tells us the connection was successful.

4.3 Receive ADC Voltage Values:

For taking the values we use CONFIG_ and getraw_functions to enable the ADC readings.They are basically voltage readings(raw values) which tell us the movement of the joystick.They range from 0-4095,as ADC works in 12 bit it has 4096 (2 raise to 12).We get largest and smallest values for the extreme movement according to which we can design the task.

4.4:Bluetooth Joystick:

We integrate the bluetooth connection and the values returned to form a combined system of a working bluetooth joystick which can be used in a game for an appropriate task which requires 2-D movement.We integrate them in the same code and according to range of ADC values we give the required task to send the mouse and keyboard values according to respective component.

On moving the joystick we can assign the mouse to go up or down according to our necessity

5. PROJECT CONCLUSION AND FUTURE WORK:

5.1 EFFICIENCY AND CURRENT ACCURACY:

I. Bluetooth Connection

As the aim was to use BLE in our project. When checked out at different stages of the project, Bluetooth Low Energy (BLE) was brought into the code and we were successful in detecting ESP32 as HID Mouse, HID Keyboard, HID Joystick and no problems were faced. In most of the devices, the ESP32 was displayed in the available Bluetooth devices in the Bluetooth Settings Menu. The pairing was also set up and connection was successful with almost every device.

ii. Reading Joystick Values

After mentioning the GPIO of ESP32 in the code and connecting them accordingly, we were able to get values from the joystick. It was observed that even with a slight change in connection of the Joystick Module with the ESP32, errors started to arise and impractical values were received. So it is essential to make sure the connection is alright.

The ADC values were recorded properly in both the devices used by Krishna and Om. The code will be developed further to make the most out of the joystick.

5.2 WHAT DID WE ACHIEVE? :

1. We got to learn Embedded Systems Technology, and how far we can get with it. This was just the beginning to this technology, but it seemed very interesting.

2. We got to know more about Bluetooth, its various types, its various protocols, how to code it and the numerous functions in it which can be employed together with ESP32.

3. We also got to learn about FreeRTOS and how different processes work in it.

4. We got to explore the capabilities of ESP32 and what magic it can do. From learning about it pins to ESP-IDF, we tried to make ourselves comfortable with the ESP-IDF development environment. And also got into the depth of its various functions.

5.3 FUTURE ASPECTS OF PROJECT:-

Since the gaming industry is growing at such an enormous rate, we plan to maximise the functions our Joystick offer to keep up with the growth. We plan to add more Joystick modules and more buttons to make it compatible with almost every device and can support almost every game available.

Also, we can use this Joystick in other sectors to control various equipments or some other things which require normal remotes. We are still exposing ourselves to various domains, so we can use our further knowledge to bring something unique out of this Joystick and not just limit its use to few sectors.

6.REFERENCES:

ESP-32-WROOM32:
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

ADC:

https://github.com/espressif/esp-idf/tree/master/examples/peripherals/adc/single_read/single_read

Bluetooth-Mouse:
https://github.com/asterics/esp32_mouse_keyboard/blob/master/main/ble_hidd_demo_main.c

Bluetooth Joystick:

https://github.com/wolfeidau/esp32-hid-joystick

Networking:

https://www.techtarget.com/searchnetworking/definition/networking

ESP32(Services):

https://atadiat.com/en/e-all-about-esp32-part-4/

Esp gatt Services :
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/esp_gap_ble.html

Protocol:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/protocols/esp_local_ctrl.html

Arduino Reference: https://github.com/T-vK/ESP32-BLE-Mouse