#### **PC Game Controller**

An

Final project report submitted

By

**SEEEMA BHAGANAGRE**

**(**[**bhaga001@cougar.csusm.edu**](mailto:bhaga001@cougar.csusm.edu) **- 200027521)**

**Under the esteemed guidance of**

**Dr. Ali Ahmadinia**

**Course Instructor for Internet of Things**



**California State University, San Marcos**

**California**

**2020**

**Abstract:**

“Internet of things could be defined as a technology the interconnection of devices both digital and computing devices via the internet, which has the capacity to sending and receiving data over the internet to other connected devices without any human or human-computer interaction.” The goal of the project is to implement a game controller for a PC. But unlike the definition of IoT my final project of this course needed a little human interaction to play a game with the help of two boards provided, namely, the STM32 NUCLEO-F401RE and the mems motion environmental sensor board the X-NUCLEO-IKS01A2. The game controller used to play the famous Google “T-rex” game. It is also commonly called as the ‘offline game’. This game is played with the help of just a single button from the keyboard that is the ‘space bar’. So, every time the player presses, he spaces bar, it made the ‘Dino’ or ‘T-rex’ jump over a huddle. This project aimed to develop a game controller for the same game with the given two boards. So, every time the player pressed the user button (blue) on the basic board or moved the motion memes board which is equipped with a gyro-meter in it, it needs to make the ‘T-rex’ jump. The board was connected to the Pc with the help of a USB cable and the was coded in embed. The user’s input was sent to a script written in python with the help of a serial communication established between the embed code and the script. These values were then input to the bord and the script and the output was seen on a computer screen that is being able to play the online game.

**Introduction:**

An Embedded system, like the name, says is a composed of both the computer software as well as the computer hardware like the processor, memory and I/O devices that perform certain specified actions. They are also termed as real-time control systems. They could either be used as standalone or could be used a part of a huge system. They are built-in with sensors that take the input from the environment they are a part of and then convert this input to a signal, which could either be digital or analogue.

On the other hand, embed is a platform which as a large collection of software and hardware libraries. It is an open-source online platform which could be used in the developing smaller projects based on hardware and the software such as the Internet of Things project.

**Aim:**

The main aim of this project was to make a PC game controlled with the STM32 NUCLEO-F401RE board and the motion mems environmental X-NUCLEO-IKS01A2 sensor board.

**Objective:**

The objective of this project is to use a NUCLEO-F401RE board with the sensor’s extension board X-NUCLEO-IKS01A2 as a game controller for a PC. The Nucleo board and the sensors that collect the user input and send it to the PC with the USB cable and a script (run on the pc) that read the data sent by the board and simulate the corresponding input on the pc.

**Requirements of the Project:**

The hardware and the software requirements of this project are,

* Hardware Requirements: STM32 NUCLEO-F401RE board, X-NUCLEO-IKS01A2 (motion mems environmental sensor board), a USB cable, working laptop or desktop.
* Software Requirements: Python (with PIP commands), Hercules (to check the serial connection), mbed complier, and an online game.

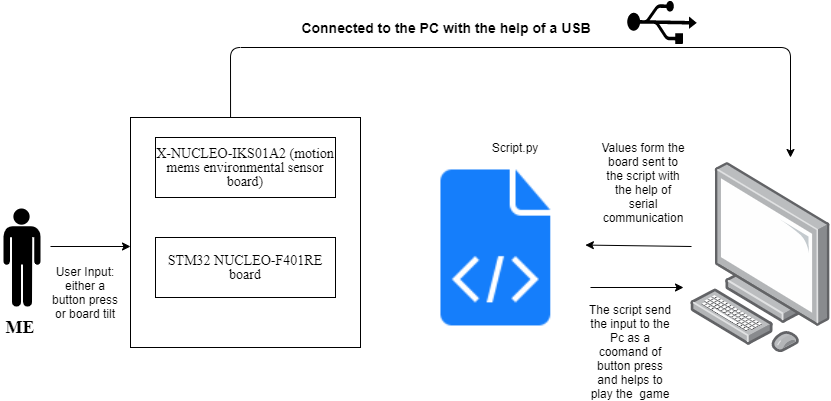
**Specification of activity (Working of the Project):**

The specification of activities is as follows:

* Firstly, I had to configure the board with the pc with the help of a mbed code with the help of serial communication and the of a USB cable.
* It was with the help of the mbed code that would send a value to the Pc as an instruction when an action occurred. In my case, I sent ‘1’ as a value. This part of the code was built with the help of the basic Nucleo board, and every time I press the button attached on the board a value ‘1’ was sent.
* It was with the help of serial communication that I established between the board and the Pc the value was sent.
* There was a python script which included the “pyautogui” and the “pyserial” libraries with the help of which a serial object was created.
* This script would receive this value store in a variable named ‘buttonState (in my code)’ and then read this string of data sent line by line and perform the needed action (of jump) with the help of an ‘if-else condition’ based on the button being pressed.
* The ‘pyautogui’ is the command used to play the game.
* The second part of this project is to play the same game with the senor board attached on top of the baseboard. That is every time I tilted the board upwards the ‘Dino’ had to jump.
* The senor board is equipped with a three-dimensional accelerometer, three-dimensional gyroscope, three-dimensional magnetometer, humidity, temperature, and pressure sensors. This board fits right on top of the baseboard.
* I used the gyroscope on this board to send input to the Pc. Since the value of movement, if recorded in six dimensions, I first had to move the board in all direction to known if my code worked right. And then I had to move in one particular motion and in one specific direction to detect which of the horizontal and vertical values were recorded on the movement of the board.
* The co-ordinates, ‘Xh’ and ‘Zh’ were used for the tilting motion of upwards and downwards respectively.
* Then based on mbed code, I was able to send the input value to the Pc.
* Finally, with the help of the button on the baseboard and the sensor board, I managed to play a game with the same script running on the background.

**Conceptual design of the system:**

The figure below shows the basic overview of the basic flow of data. It is the user that gives the input by either pressing the button or the moving the board in one particular direction. The value of either button press or the motion sensors value are sent to the pc with the help of the USB, here serial communication is used. The data that is sent to the PC via a python script which takes these input values and then helps them convert into a similar command of a button and then we should be able to play the game with the help of that.



**Figure: top-level-flow-chart (system specification)**

* **Detailed design and implementation of the code:**

Implementation of the code:

* The part the of the code which makes the T-rex jump when the gyroscope is tilted upwards.

if (GyroMoveFlag) //If motion detected

{

GyroMoveFlag = 0;

LSM6DSL\_Event\_Status\_t GyroStatus;

Gyro->get\_event\_status(&GyroStatus); //Get the gyro status

if (GyroStatus.D6DOrientationStatus) //if move is within 6D

{

send\_TiltUPValue(); //check gyro motion and send 1 if tilt motion is upward

}

}

Here, when a motion is detected in the gyroscope, the flag which was initially one is now set to zero, then the status of the gyroscope is tested, if there is a change in the status of the gyroscope, then the T-rex jumps of it is tilted upwards (Xh=1) or and it does not jump if the board od moved downwards (Zh=1).

* The part the of the code which makes the T-rex jump when the user button is pressed and the led glows.

ButtonState = mybutton; //Read button status

if(ButtonState==Pressed && OldButtonState ==Released) //if button is pressed and old status was released

{

pc.printf("1"); //send 1 to the PC

userLed = 1; //Turn LED on

}

else if(ButtonState==Released && OldButtonState ==Pressed) //if button is released and previous status was pressed

{

userLed = 0; //Turn the LED off

}

OldButtonState = ButtonState; //Save the old button status

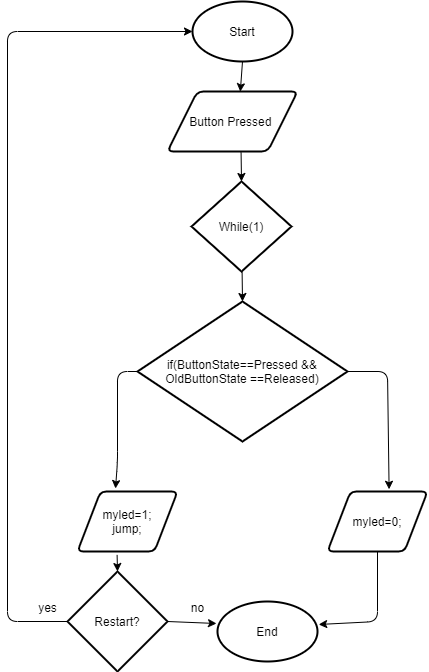
wait(0.001);

}

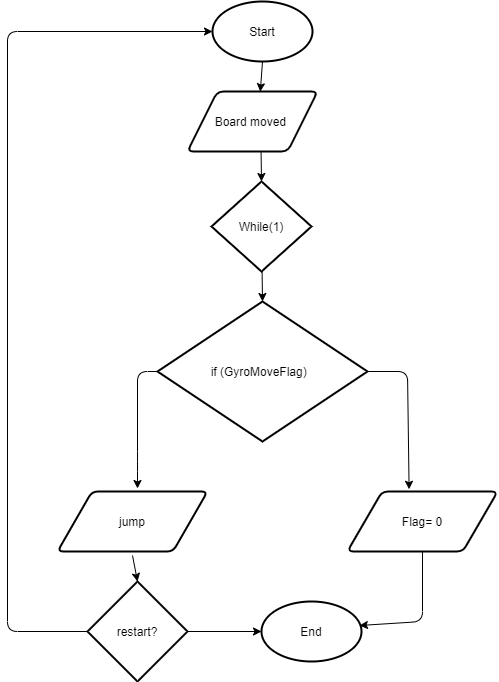
Initially the ButtonState of the user button ideal hence the value one, later once the button is pressed and the value is check for the old and new values of the button and based on that value one is sent to the serial object which helps the T-rex jump and the led also glows.

The flow charts for the project are as follows:

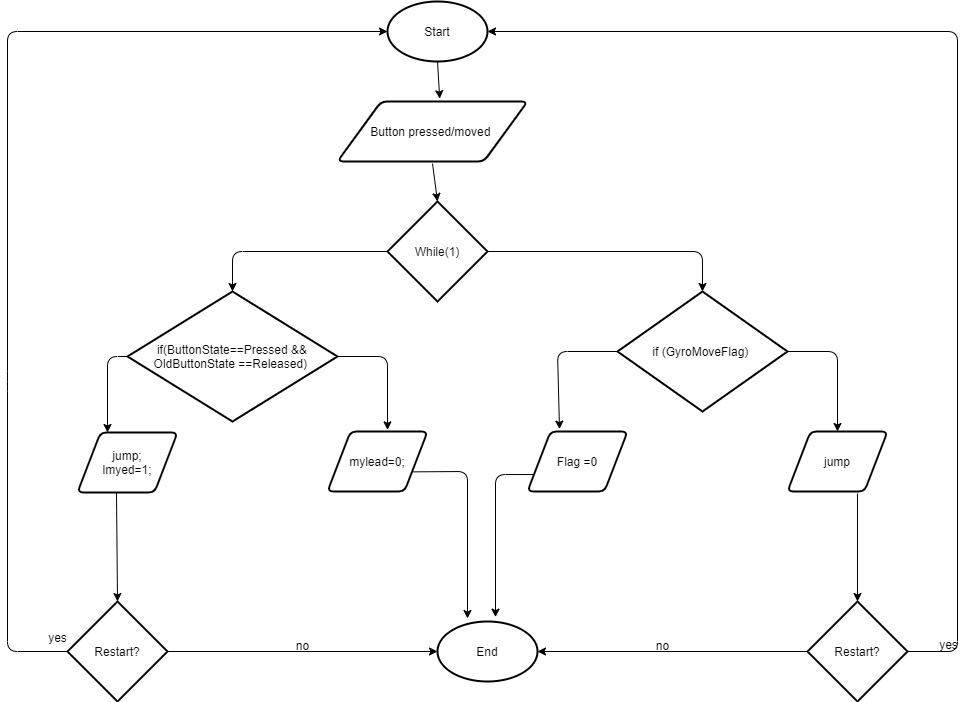
* Flowchart when the user button on the Nucleo board is pressed



* Flowchart when the board is moved and a motion is detected by the gyroscope



* Flowchart when the board is moved and also the user button is pressed.



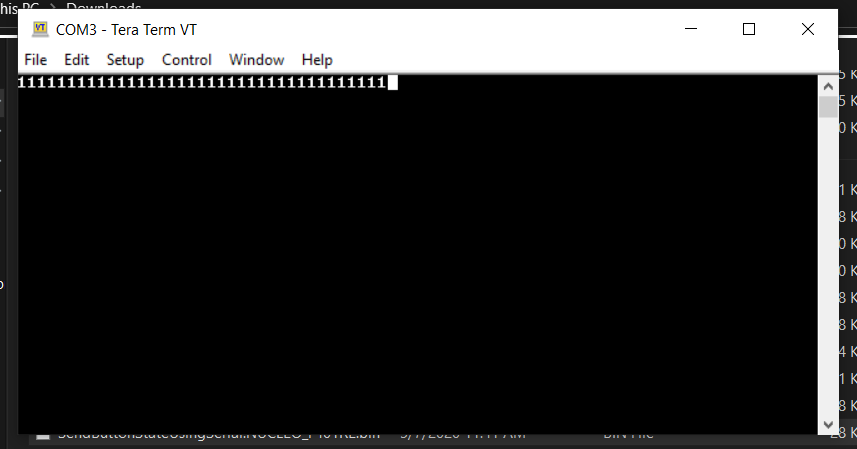
**Discuss the main challenges of implementation:**

The major challenges while implementing this project were,

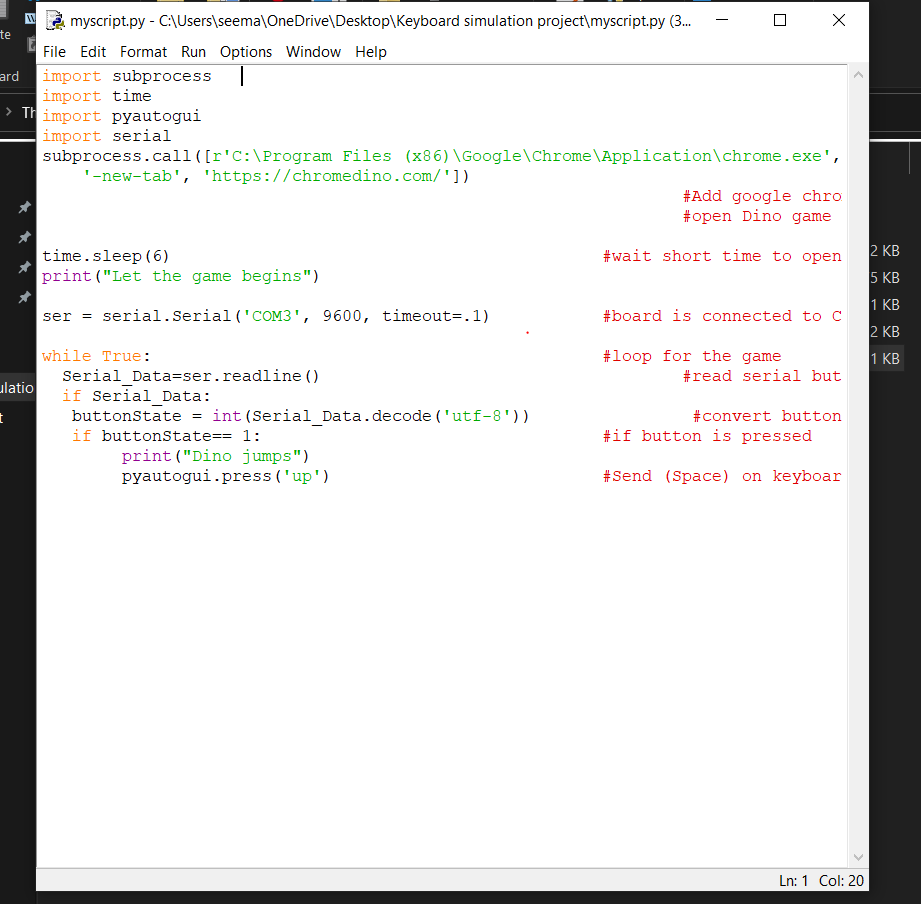
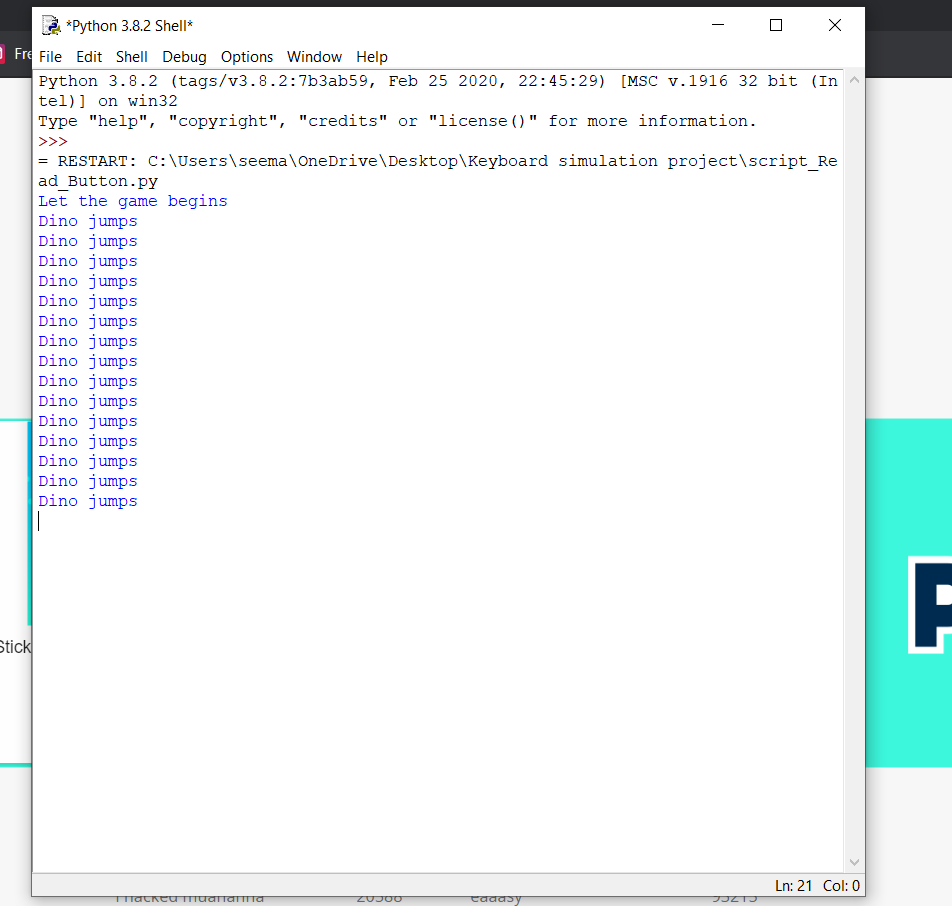
* Configuring the board as a keyboard,
* Find proper lead or help from the internet to get started with,
* To write a python script, since I had to learn and understand to run the clip, had to install pip commands to include python libraries.
* Writing code for the gyroscope, we’ve done labs with the accelerometer, temperature, humidity and pressure sensors, but making the gyroscope work and the understand the coordinates was challenging.

**Testing:**

* Step 1: Open a browser of your choice and google search “[6DOrientation\_IKS01A2](https://os.mbed.com/teams/ST/code/6DOrientation_IKS01A2/) in mbed” it opens into a new page and then import that into the embed compiler.
* Step 2: Give a program of your choice and press ‘Import’
* Step 3: Open the main.cpp and copy-paste my code that is included in my zip file with the ‘main\_cpp’.
* Step 4: Now, press on the compile button on the top menu bar,
* Step 5: You will see that the code is successfully compiled and ‘your\_program\_name.bin’ gets downloaded.
* Step 6: Now you need to flash this .bin file on your flash drive,
* Step 7: To check if the serial communication is properly established and if the values are being sent, you could check that on ‘Tera Term’. The values would look something like this.



* Step 8: For this step, you need to have python installed on your computer. You also need to install the ‘pip command’. Then you also need to install the ‘pyautogui’ and ‘pyserial’ libraries. These could be installed only after installing the pip command. Also, make sure you update you browsers location on your local pc in the script.
* Step 9: You could directly run the ‘script.py’ file, a python script. To run it you could just double tap on the file or you could right-click and open with an IDLE.
* Step 10: the menu bar on the top has a run option, select that, a dropdown appears, select that run module option. This could open a new page and then redirects to your browsers.
* Step 11: Now you be able to play the game either with a button press or just tilting the board or both.
* The screen shot of the running script looks something like this,

**Summary:**

Video link: <https://drive.google.com/file/d/1old-cgXbHsC5xtpeO_9SJV62h7RSYR05/view>

**Reference/Bibliography:**

* <https://os.mbed.com/teams/ST/code/6DOrientation_IKS01A2/>
* <https://www.instructables.com/id/Play-Dinosaur-Game-Using-Arduino-and-Python3/>
* <https://www.youtube.com/watch?v=F2TZuOhsjx4&feature=emb_logo>
* <https://www.youtube.com/watch?time_continue=1&v=5ZsYP45X9Qc&feature=emb_loo>
* <https://create.arduino.cc/projecthub/rafitc/automated-dino-game-using-arduino-4fb380>
* https://www.youtube.com/watch?v=zPMr0lEMqpo