18-10-2022

INTERFACING PI PICO WITH ADXL AND MPU6050 TO RUN ON EDGE IMPULSE

RESET THE BOARD

- To reset the board, press the boot button on the board and connect to PC.
- After accessing the bootloader FS copy paste the “***flash_nuke.uf2”*** into the filesystem of Pico***.***

CONNECTING THE SENSORS TO PICO

- Follow the below pinout instructions to connect the sensor to the board
- (“make sure to check the voltage level of the output as some sensors wont work with 5v and might need level shifter or other GPIO”)

PINOUT for MPU6050 with Pico

VCC - GPIO 36 (3v3 out)
GND- GPIO 38
SDA – GPIO 06
SCL – GPIO 07

PINOUT for ADXL with PICO

VCC – GPIO 40
GND – GPIO 38
X-OUT – GPIO 31
Y-OUT – GPIO 30
Z-OUT – GPIO 34

INSTALL ARDUINO LIBRARIES

- Go to Library Manager and Install
- Raspberry pi Pico/rp2040 by Earle F.Philhower, III Version 2.5.4 (install only this version as it is stable and works with edge impulse)
- Install the Adafruit MPU 6050 library

TEST UPLOAD WITH SAMPLE BLINK

Upload a sample blink sketch found in examples and check upload status

UPLOAD THE DATA FORWARDER CODE FOR PICO WITH MPU OR ADXL

Find the code on the GitHub repo or if writing custom code interface, the sensor and make sure to print all the sensor values separated by commas on the serial monitor and print nothing else as the edge impulse data forwarder in cli will read the serial port where we can assign names to these values.

Also check for movement output on the serial plotter for different movements and there should be respective change in sensor values.

Close the serial monitor !! as edge impulse will read from the serial port and will not be able to Arduino ide serial monitor is open

OPEN EDGE IMPULSE DASHBOARD

- Login into your account and create a new impulse
- Go under the devices section

FORWARD SENSOR DATA FOR EDGE IMPULSE DATA- ACQUISITION

- Go to the Edge Impulse CLI and run ***$ edge-impulse-data-forwarder –clean***
  If running for the first time.
- Go ahead and login with ID and Password after it detects the device and the sensor axis output it will ask to name these values by comma separation
  For MPU name them ***Ax, Ay, Az, Gx, Gy, Gz*** and for ADXL name them ***Ax, Ay, AZ***
  These will be the sensor values on edge impulse.
- Additional cli commands for data forwarder
  - -- frequency 100 (override frequency)
  - –baud rate 115200 (override Baud Rate)
- Check connection in edge impulse under Devices

DATA ACQUISITION AND BUILDING MODEL

Use the connected board to sample the data and train your model according to the preprocessing blocks

MOEDL DEPLOYMENT

- After building the model deploy it as ARDUINO LIBRARY and download the .zip file
- Open Arduino Ide and under library manager add the downloaded file as library
- Open the continuous buffer code uploaded on Github
- Change the Header file import to the downloaded edge impulse library
- Upload the code to Pico and make sure the file gets copied onto the file system and the com port is reset if not working flash the Nuke.uf2 file again.

CUSTOM CODE

To write custom code go to the example section and under edge impulse downloaded library will be a static_buffer.ino file

- We need to convert this to a continuous buffer add the sensor code from the data forwarder to static buffer and put the custom sensor values in the feature array until its full and run the prediction functions and test the output.

RUN IMPULSE

- The impulse prediction output can be seen in the serial monitor but to see the raw output and more debug options use the edge impulse cli tool
  - Run ***$ edge-impulse-run-impulse***
- Additional Commands:
  - ***-- raw*** (raw data output)!
  - ***--debug*** (debug mode and options)
  - ***--continuous*** (run the impulse continuously)
- In Arduino IDE go to preferences and ***select show verbose output during compile*** **and** ***upload and uncheck verify code after upload***