dsPIC33CK Gesture Recognition

Objective:

This manual offers a detailed walkthrough for constructing a Gesture Recognition machine learning model using MPLAB ML Development Suite for dsPIC33CK Digital Signal Controller (DSC). It outlines the process of data collection, transmission to the ML Model Builder, creation of a customized Gesture Recognition model for precise data classification, and deployment onto the DSC.

Although it focuses on gesture recognition, the techniques it covers can be useful in many areas where classifying time series sensor data is important, like predicting maintenance, activity recognition, sound classification, and keyword spotting.

Hardware:

- DSPIC33CK Curiosity Development Board

- BMI160 6DOF IMU 2CLICK

Software Tools

- MPLAB X IDE (Integrated Development Environment) (v6.15 or later)
- XC16 2.1 Compiler
- MPLAB Data Visualizer
- MPLAB Development Suite
    - ML Data Collector
    - ML Model Builder

Board Connection

The IMU 2 click board is plugged into the development board's socket A mikroBUS™. Make sure to verify the pin labels to ensure correct connections between the boards.

Before getting started, make sure to install all the necessary software tools. We'll be using MPLAB X IDE and compiling the project with the MPLAB XC16 compiler. You can find both MPLAB Data Visualizer and the Machine Learning Plugin in the plugin update center within MPLAB X IDE. You will also need to sign up for myMicrochip account to access MPLAB X ML Development Suite

Data Collection

The initial stage of constructing a machine-learning model involves gathering data, which serves as the foundation for training the ML model to distinguish various types of gestures. Specifically, in this instance, the model will undergo training to recognize distinct gestures such as idle (indicating no movement), up-down (indicating vertical movement), wave (indicating horizontal movement from left to right), and wheel (indicating clockwise rotation).

Exercise Files:

- The firmware and dataset used in this tutorial and other MPLAB X project files can be found in this GitHub release.
- For those interested in the IMU data collection firmware and usage instructions for the dsPIC33CK Curiosity, they are available for download from the DSPIC33C IMU Data Logger.

Data Collection Overview

Before you begin gathering data samples, make sure to plan out the data collection process carefully. Remember, the quality of the collected data will directly affect how well the Machine Learning model performs.

Data Collection: Sensor Configuration

For this project, an accelerometer-only configuration was selected, along with a 100 Hz sample rate and a 16G accelerometer range. This choice was made because it was anticipated that the chosen gestures would be mostly unaffected by device rotations. Despite the gestures typically falling within a frequency range of <5 Hz (i.e., below the 10 Hz Nyquist rate), a sample rate of 100 Hz was chosen to allow for flexibility in the data collection process. The least sensitive setting of the 16G accelerometer range was chosen because micro-movements were not of interest.

Data Collection: Collection Protocol

To proceed with the data collection process, the next essential step involves creating a protocol that outlines the necessary procedures for collecting data. This entails determining the number of samples to be collected, the metadata parameters to be captured, and other relevant experimental parameters that will govern the data collection procedure. The protocol will serve as a guide for executing the data collection process and ensure that the data collected is consistent and accurate.

Data Collection: Executing the Gesture

In this project, the following guidelines dictate the performance of gestures by the test subjects.

- Subject performs gestures that follow the specified trajectory description (e.g., clockwise rotation).
- Subject performs gestures *smoothly*, in a way that feels *natural* to them.
- Subject performs gesture continuously for at least *ten* seconds.
- Subject uses hand dominant.
- Subject is sitting.
- Subject holds the board as shown in the figure below.

Data Collection: Metadata

Initially, it is imperative to address metadata, as it significantly contributes to contextualizing our data. In this application, our metadata comprises solely the dataset, partitioned into training and testing subsets.

Data Collection: Sampling Method

At this stage, it is necessary to decide the data sampling process for our application. This involves determining the number of samples to be collected and defining the necessary steps for measuring the data. In the context of this example application, the methodology can be summarized in the following steps

Record and label segments for each of the following gesture settings: idle, up-down, wave, and wheel.

How to Configure, Compile, and Flash

This document explains the steps involved in configuring the data logger firmware build, compiling it, and flashing it to the DSPIC33C DSC device. Follow the instructions below:

1. Connect the DSPIC33CK Curiosity board to your PC using a USB cable.

2. Install the MPLAB X IDE and XC16 compiler on your PC. These are essential tools for loading the data logger project and programming the DSPIC33C DSC device.

3. Open the firmware/dspic33ck-curiosity-imu-data-logger.X project folder in MPLABX

4. Select SNSR_TYPE_BM160 for MPLABX Project Configuration

5. Make sure the DATA_VISUALIZER_BUILD macro is set to TRUE in firmware/dspic33ck-curiosity-imu-data-logger.X /app_config.h

6. In the same app_cofig.h file, modify as desired the high-level sensor parameters like sample rate (SNSR_SAMPLE_RATE), and others by changing the macro values defined.

When you are satisfied with your configuration, click the Make and Program Device icon button in the toolbar (see image below for reference).

Figure 1-1 Programming the Device

Firmware Operation

The data streamer firmware will output sensor data over the UART port with the following UART settings:

- Baud rate 38400
- Data bits 8
- Stop bits 1
- Parity None

Usage with the MPLAB Data Visualizer and Machine Learning Plugins

This project can be used to generate firmware for streaming data to the MPLAB Data Visualizer plugin. When the firmware is flashed, follow the steps below to set up Data Visualizer.

1. Connect the Curiosity board to your computer, open up MPLAB X, then open the Data Visualizer plugin.

2. Click the Load Workspace button as highlighted in the image below. Select mplab-data-visualizer-workspace-accel-3-axes.json workspace file included in this repository. You can always modify the configuration when it is loaded if needed.

Figure 1-2 Loading .json file to variable streamer

3. Next, select the Serial/CDC Connection that corresponds to the Curiosity board, adjust the baud rate to 38400, then click **Apply**.

Figure 1 -3 Configuring the COM Port setting

4. Use the play button on the Serial/CDC Connection to connect to the serial port. When the connection is made, the DSPIC33C-DSC is available for use with the variable streamer.

5. Switch to the **Variable Streamers** tab, then select the aforementioned Serial/CDC Connection from the dropdown menu as the input data source for the IMU variable streamer (see image below for reference); this will enable parsing of the DSPIC33C DSC data stream. You may delete or add variables here if your sensor configuration differs from the pre-loaded ones.

6. The IMU data is now available in the time plot. Double-click anywhere within the time plot to start/stop scrolling of the time axis.

7. Click the up-down arrow button to auto-set the time scale.

8. Click the mark icon button, then the snapshot icon button to snip the part of the time plot as data to save.

Figure 1-4 Capturing Sensor Data

Visit the *Machine Learning Plugin* page to learn more about using the Data Visualizer plugin to export your data for machine learning applications.

Model Development

Once the initial dataset for your application has been gathered, the next step is to proceed to the ML Model Builder to create your classifier model. To generate your classifier model, proceed to MPLAB ML Development Suite.

The process for accessing the Microchip ML Development Suite requires the following steps:

- Launch the MPLAB X IDE software.
- Next, launch the ML Model Builder tool.
- When the ML Model Builder page loads, you will be prompted to authenticate your account by entering your myMicrochip login credentials.

Figure 1-5 ML Model Builder login credentials

Upon successful login, users can create a new project by selecting the "CREATE PROJECT" tab, which prompts a pop-up window to specify a project name. To access a created project, simply click the "Open Project" icon button located before the project name.

Figure 1-6 Creating a new ML Project

To create new labels and metadata fields for a project, follow these steps:

1. Navigate to the Project Summary page.

2. Access Project Settings.

3. Locate the "Segment Labels" and "Metadata" fields.

4. Select the "Add Label" and "Add Metadata" tabs positioned in the top right corner of

   the page.

5. Input the necessary details for the new label and metadata fields.

6. Save the information to generate the new label and metadata field.

7. Repeat steps 4-6 for each additional label or metadata field desired.

Figure 1-7 Configuring Project settings.

Next, to import the necessary data files for the project, follow these steps:

1. Navigate to the Data Manager tab.

2. Locate the IMPORT CAPTURE FILE button, then click it.

3. Ensure that the data files to be imported are in either CSV or Wav format.

4. Select all the files required for the project.

5. Begin the import process.

Upon completion of the import process, the imported data files will be ready for utilization within the project. It is crucial to confirm that all data files are correctly formatted and labeled to guarantee precise analysis and interpretation of project outcomes.

To access the imported data files, click the icons corresponding to the desired file under the OPEN column.

Figure 1-8. Creating segments inside data file.

Once the desired file is open, navigate to the **CREATE** tab to begin segmenting the data. To create segments, drag the segments' axis to the desired locations within the data.

It is, then, recommended to correctly categorize these segments using appropriate labels and metadata fields for ease of organization and future analysis.

Figure 1-9. Labelling the segments.

To proceed, navigate to the Prepare Data view. From there, locate and click the **CREATE QUERY** button to initiate the data preparation process.

Figure 1-10. Creating a new project query.

1. Fill out the block as displayed below.

2. Press the **CACHE** button to initiate the caching process.

3. Wait for the cache to build out completely.

4. Click the **SAVE** button when the cache is fully built.

Following these steps will result in the block saving cached information. Once the caching process is complete, details such as Segment Length Distribution, Label Distribution by segment, and Label Distribution by samples will be displayed.

Figure 1-35. Fill in the details for the query.

To proceed with building the pipeline, navigate to the Build Model module and, then locate the Create Pipeline tab

Figure 1-11. Creating a new pipeline.

After clicking **CREATE NEW PIPELINE**, information regarding Pipeline displays. Click **NEXT** to proceed.

The second step in configuring the template involves selecting the appropriate pipeline based on the sensors intended for inclusion. In this instance, six signals have been gathered, comprising three from each accelerometer and gyroscope, warranting the selection of the first template.

In this step, choose the desired query for constructing the pipeline. In the provided example, "query_dspic33ck_gesture_expand" is chosen as the query and the pipeline has been named "pipeline_dspic33ck_gesture_expand". Additionally, it's essential to appropriately link sensor data signals with their respective sensor columns in the template, replicating the process demonstrated below. Upon selecting the query, ensure to click the "CREATE PIPELINE" button to preserve your selection.

Figure 1-12. Naming the new Pipeline and selecting the query.

The pipeline will be created showing parameters that will be used while building the Model. Click **NEXT**.

To begin, access the pipeline settings by selecting the edit icon adjacent to it. By default, the system employs AutoML mode. However, users have the option to utilize custom training mode if necessary. For this demonstration, we will proceed with AutoML. Adjustments will be made solely to the search settings, specifically the number of iterations and population size, as indicated below

Access the Segmenter module, which performs segmentation of the data. Set the parameters as depicted in the figure below to ensure the appropriate segmentation of the data. After setting the parameters, click the **SAVE** button to save the changes made.

Then, open the Feature Generator module. Select the features used to extract the information from the input sensor data. In this form, you can specify the types of feature generators you would like to use, the inputs into the feature generators and any parameters the feature generators take. Select the features that are shown below and click **NEXT** after selecting the required features.

Figure 1-12. Selecting the Features for Model Building

The subsequent step involves adjusting the Outlier filter based on the dataset. In our scenario, the Outlier fraction was configured to 0.05, utilizing isolation forest filtering.

Figure 1-45. Setting the Validation Parameters.

Lastly, click the **Validation** tab. This is a validation scheme that splits the data set into training, validation and (optionally) test sets based on the parameters. Select the appropriate validation method based on the project requirements. When the validation method is chosen, specify the “Validation Method”, the “Number Of Folds” and the “Test Size” to be used for the validation

process. After providing these inputs, click the **SAVE** button to save the changes made to the validation module.

Figure 1-13. Setting the Validation Parameters.

Upon configuring all settings, activate the optimization process by selecting the "Run Pipeline" button located at the top.

Figure 1-14. Starting the AutoML optimization process.

The AutoML techniques will automatically select the best features and machine learning algorithm for the gesture classification task based on the input data you provided. Note that this optimization process typically takes several minutes to complete.

After completing the process of Model Optimization, proceed to the Explore Model view. Within this view, the top five models are ranked and displayed based on their ability to produce optimal outcomes. To access additional information regarding each model, click the open model icon that is situated on the left-hand side of its respective name.

Figure 1-15. Exploring the model.

Users can access different output results via the options available at the top. These results include confusion matrices, graphs, and flowcharts, which can be explored for detailed analysis through modules such as confusion matrix, feature embedding, feature visualization, and different summaries.

Figure 1-48. Different modules under the explore model.

After confirming satisfactory model results, users can navigate to the Test Model view. To filter data and isolate test samples, follow these steps:

1. Choose the desired Session.
2. Click the upside-down triangle icon in the Fold column.

Figure 1-16. Testing the Model.

To compute accuracy and generate a confusion matrix for the test samples, follow these steps:

1. Click the ellipsis (…) icon in the left-most column of the table.

2. Select "Select All" from the dropdown menu to include all test samples.

3. Click the "Recognize" button to compute accuracy metrics.

4. Once accuracy is computed, click "Summarize" to generate the confusion matrix.

5. Note that this process may take a few minutes. Upon completion, a table summarizing.

classification results will be presented.

Figure 1-17. Confusion matrix summary.

SUMMARY

	Horizontal	Stationary	Unknown	Vertical	UNK	Ground Truth	Support	Sensitivity(%)
Horizontal	237	0	0	0		4	237	100.00
Stationary	1	207	0	0		4	208	99.52
Unknown	0	0	0	0		0	0	
Vertical	0	0	0	127		5	127	100.00
Predicted	238	207	0	127		13	572	
Pos_Predic(%)	99.58	100.00		100.00			Acc(%)	99.83

To deploy your model, follow these steps:

1. Navigate to the Download Model tab.

2. Fill out the Knowledge Pack settings using the Pipeline, Model and Data Source you created in the previous steps.
3. Select the library output format.

4. Click the **Download** button.

Figure 1-51. Downloading the model.

By completing these steps, you will be able to deploy your model and make it available for use. It is important to ensure that all necessary settings are correctly entered before proceeding with the download.

Figure18. Selecting the parameters for downloading the model.

Knowledge Pack Integration

Integrate the Knowledge Pack APIs into your embedded device firmware by following the Knowledge Pack Integration.

DSPIC33CK Curiosity Gesture Recognition Firmware Overview

For a description of the demo firmware included with this project including operation, usage and benchmarks, see the README file in the GitHub Repository.

Final Remarks

Upon completing this guide, you will possess a foundational knowledge of developing a Smartbell application utilizing MPLAB ML Development Suite in conjunction with the DSPIC33CK Curiosity Board and BMI160 6DOF IMU sensor. For additional insights into machine learning capabilities at Microchip, please refer to the resources available on Artificial Learning and Machine Learning.